TE
662
.A3
no.
FHWA-
RD-
76-68

Report No. FHWA-RD-76-68

VEHICLE DETECTION--PHASE III
PASSIVE BUS DETECTOR / INTERSECTION PRIORITY SYSTEM DEVELOPMENT

Vol. II, Part II. Appendix D--Computer Program Listings

Dept. of Transportation
MAR 10 1977
Library

December 1975
Final Report

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

Prepared for
FEDERAL HIGHWAY ADMINISTRATION
Offices of Research & Development
Washington, D.C. 20590

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The contents of this report reflect the views of the contracting organization, which is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policy of the Department of Transportation. This report does not constitute a standard, specification, or regulation.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein only because they are considered essential to the object of this document.

TE
662
.A3
no.
FHWA-
RD-
76-68

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-RD-76-68		

4. Title and Subtitle	5. Report Date
VEHICLE DETECTION - PHASE III PASSIVE BUS DETECTOR/INTERSECTION PRIORITY SYSTEM DEVELOPMENT Volume II, Part II: Computer Program Listings	December 1975
	6. Performing Organization Code
7. Author(s) P. Anderson, R. Larson, R. Lubke, G. Putnam and D. Wick Printouts	8. Performing Organization Report No. 2551-40274
9. Performing Organization Name and Address Honeywell Inc. 2600 Ridgway Parkway Minneapolis, MN. 55413	10. Work Unit No. (TRAIS) FCP-32L1012 11. Contract or Grant No. DOT-FH-11-8149
12. Sponsoring Agency Name and Address Federal Highway Administration Office of Research and Development Washington, D.C. 20590	13. Type of Report and Period Covered Final report 14. Sponsoring Agency Code T-0188

15. Supplementary Notes

16. Abstract

This volume contains the description of the software developed for the Passive Bus Detector/Intersection Priority System. The program documentation is formatted according to the Program Documentation section of the FHWA Documentation Standards. The computer program documentation is referenced in other volumes of this report as the following appendices:

Appendix A. FORTRAN IV Version of the Vehicle Classification Algorithm
Appendix B. BPCU Program for Minneapolis Installation
Appendix C. BPCU Program for Washington Installation

Appendix C is to be treated as an addendum to Appendix B and contains documentation of only those modules which are different from the Minneapolis Installation program in Appendix B.

Appendix D, Software and Computer Printouts, is presented separately as Part II of this volume.

Dept. of Transportation
MAR 10 1977
Library

17. Key Words	18. Distribution Statement
Passive bus detector, ILD signal processing, Microprocessor, Fully independent controller	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	409	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

APPENDIX D. PROGRAM LISTINGS

This appendix contains the program listings for three computer programs.

FORTRAN IV Version of the Vehicle Classification Algorithm

Compilation listing for program execution on the XDS Sigma 5.

PBD33J - BPCU Program for Minneapolis Installation

Assembly listing of the Passive Bus Detector Software in the microcomputer of the Bus Preempt Controller Unit (BPCU) as field tested at 33rd Avenue and Johnson Street Northeast in Minneapolis, Minnesota.

PBD14C - BPCU Program for Washington Installation

Assembly listing of the Passive Bus Detector Software in the microcomputer of the Bus Preempt Control Unit (BPCU) as field tested at 14th and C Streets in Washington, D. C.

PROGRAM LISTING OF THE FORTRAN IV VERSION
OF THE VEHICLE CLASSIFICATION ALGORITHM.

```
19147 NOV 23,1976 ID=000B-C00
JOB =510, PBOVE
ASSIGN M:S1,(DEVICE,CRA01),(FBCD)
ASSIGN F:5,(DEVICE,CRA03),(FBCD)
ASSIGN F:6,(DEVICE,LPA02)
ASSIGN M:B9,(FILE,MAIN)
FORTRAN S1,B9,LS
```

APPENDIX D. FIGURE A.8

PROGRAM LISTING OF THE FORTRAN IV VERSION OF THE VEHICLE CLASSIFICATION ALGORITHM

```
C*****************************************************************
C*********************  FORTRAN VEHICLE CLASSIFIER  ****************
C*****************************************************************
C*************************  MAIN PROGRAM  ***********************
C*****************************************************************
C  THE PURPOSE OF THE MAIN PROGRAM IS TO READ A VEHICLE SIGNATURE AND
C CALL THE VEHICLE CLASSIFICATION ROUTINE TO EVALUATE IT. THE MAIN
C PROGRAM ALSO READS IN THE PRINTOUT CONTROL PARAMETER, CLEARS THE PERR
C AND VCLOG ARRAY, AND JUST PRIOR TO PROGRAM TERMINATION, PRINTS OUT THE
C PERR AND VCLOG ARRAYS.
C*****************************************************************
C******  SUBROUTINES CALLED BY THE MAIN PROGRAM  *****************
C  VCLAS-CLASSIFYS A VEHICLE  SIGNATURE AS BUS OR NON-BUS
C*****************************************************************
C*********************  VARIABLES  *******************************
C PERR-PREEMPT ERROR BUFFER
C SDB-SIGNATURE DATA BASE
C VCDT-VEHICLE DETECTOR INDEX NUMBER
C VCLOG-VEHICLE CLASSIFIER LOG
C VCOUT-PRINTOUT CONTROL PARAMETER
C*****************************************************************
      COMMON VCOUT,PERR(17)
      COMMON VCDB(25),VCLOG(15),VCCO(10),SDB(32,8),VCDT
      INTEGER VCOUT,PERR
      INTEGER VCDB,VCLOG,VCCO,SDB,VCDT
C
C     VCOUT CONTROLS PRINTOUT IN THE VEHICLE CLASSIFICATION ROUTINE
C             0  =  PRINTOUT OF SUMMARY ONLY
C             1  =  LIMITED PRINTOUT
C             2  =  FULL PRINTOUT
      READ (5,98) VCOUT
      VCOUT = VCOUT - 1
C             CLEAR VCLOG ARRAY
              DO 4 I=1,15
              VCLOG(I) = 0
   04         CONTINUE
C             CLEAR PERR ARRAY
              DO 6 I = 1,17
              PERR(I) = 0
   06         CONTINUE
C     SPECIFY DETECTOR
C        ASSUME ALL SIGNATURES FROM DETECTOR ONE
      VCDT = 1
C
C     READ IN VEHICLE SIGNATURE DATA
   10 READ(5,99) (SDB(VCDT,I),I=1,7)
      IF (SDB(1,1).EQ.0) GO TO 20
C     ADD END OF SIGNATURE FLAG TO SIGNATURE DATA BLOCK
      SDB(VCDT,8) = 1
C
C     CALL VEHICLE CLASSIFIER
      CALL VCLAS
      GO TO 10
   20 WRITE(6,95)
      WRITE(6,94)(VCLOG(J),J=1,6)
      WRITE(6,96)(PERR(J),J=1,17)
   23 STOP
```

```
 94  FORMAT(10X,36HVEHICLE CLASSIFICATION - VEHICLE LOG/
    +15X,20HNOISE                    ,I10/
    +15X,20HGROSS TEST FAILURES ,I10/
    +15X,20HSTOPPED VEHICLES     ,I10/
    +15X,20HMOVING VEHICLES      ,I10/
    +15X,20HNON-BUSES            ,I10/
    +15X,20HBUSES                ,I10///)
 95  FORMAT(1H1,16HEND OF FILE READ)
 96  FORMAT(///10X,25HPREEMPT ERROR BUFFER DUMP/15X,10HERROR FLAG,I10/
    +45X,20H COUNTER    KEY WORD/
    +15X,30HOVERFLOW                        ,2I10/
    +15X,30HDETECTOR OUT OF RANGE           ,2I10/
    +15X,30HAMPLITUDE VALUE OUT OF RANGE    ,2I10/
    +15X,30HTIME VALUE OUT OF RANGE         ,2I10/
    +15X,30HTIME VALUE OUT OF ORDER         ,2I10/
    +15X,30HNOISE                           ,2I10/
    +15X,30HEBS STATUS CHANGE DURING COPY   ,2I10/
    +15X,30HSIGNATURE LOSS                  ,2I10)
 97  FORMAT(1H ///16HEND OF FILE READ)
 98  FORMAT (I2)
 99  FORMAT (7I4)
     END
```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
I	I	SCALR	00000 V	1	J	I	SCALR	00001 V	1	PERR	I	ARRAY	00001 C	17
SDB	I	ARRAY	00044 C	256	VCCO	I	ARRAY	0003A C	10	VCDB	I	ARRAY	00012 C	25
VCDT	I	SCALR	00144 C	1	VCLAS		SPRAG	EXTERN		VCLOG	I	ARRAY	0002B C	15
VCOUT	I	SCALR	00000 C	1										

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
4	0000E	6	00018	10	0001D	20	00034	22	00052	94	00054
95	00091	96	00099	97	00106	98	0010E	99	00110		

LOCAL VARIABLES (2 WORDS):

00000 I 00001 J

BLANK COMMON (325 WORDS):

00000 VCOUT 00001 PERR 00012 VCDB 0002B VCLOG 0003A VCCO 00044 SDB
00144 VCDT

EXTERNAL SUBPROGRAMS REQUIRED:

VCLAS 9BCDREAD 9BCDWRIT 9ENDIOL 9INITIAL 9IODATA 9STOP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS	
GENERATED CODE:	277	00115	
CONSTANTS:	0	00000	
LOCAL VARIABLES:	2	00002	
TEMPS:	2	00002	
TOTAL PROGRAM:	281	00119	(PLUS BLANK COMMON)

```
C
      SUBROUTINE ERE(I,J1,K)
C*****************************************************************
C  THIS PROGRAM RECORDS AND PRINTOUTS INFORMATION CONCERNING ERRORS
C DETECTED BY SUBROUTINE VCLAS.
C*****************************************************************
C***************  NO SUBROUTINES CALLED BY THIS ROUTINE  **************
C*****************************************************************
      DIMENSION K(17)
      COMMON VCOUT,PERR(17)
      COMMON VCDB(25),VCLAG(15),VCCO(10),SDB(32,8),VCDT
      INTEGER VCOUT,PERR
      INTEGER VCDB,VCLAG,VCCO,SDB,VCDT
      DIMENSION CON(33)
      DATA (CON(J),J=1,33)/2.5,0.43, 7.0,0.0,0.0,0.0,2.73,0.876,2.52,0.139,
     +0.0,0.0,0.0,0.0,0.277,0.0198,0.298,0.947,0.96,1.66,1.36,2.17,4.14
     +1.444,3.93,6.47,3.0,5.0/
      IF (VCOUT)  30,25,20
   20 WRITE(9,99) I,J1
      IF(I.EQ.0) WRITE(9,900)
      IF(I.EQ.1) WRITE(9,901)
      IF(I.EQ.2) WRITE(9,902)
      IF(I.EQ.3) WRITE(9,903)
      IF(I.EQ.4) WRITE(9,904)
      IF(I.EQ.5) WRITE(9,905)
      IF(I.EQ.6) WRITE(9,906)
      IF(I.EQ.7) WRITE(9,907)
      WRITE(9,98)(VCDB(J),J=1,8),((CON(J),VCDB(J)),J=9,12),(VCDB(J),J=
     +13,14),((CON(J),VCDB(J)),J=15,16),VCDB(17),((CON(J),VCDB(J)),J=
     +18,19),(VCDB(J),J=20,22)
      GO TO 30
   25 WRITE(9,97) I,J1
   30 I1 = (I*2) + 2
      I2 = I1 + 1
      K(I1) = K(I1) + 1
      K(I2) = J1
      RETURN
  900 FORMAT(15X,8HOVERFLOW/)
  901 FORMAT(15X,21HDETECTOR OUT OF RANGE/)
  902 FORMAT(15X,28HAMPLITUDE VALUE OUT OF RANGE/)
  903 FORMAT(15X,23HTIME VALUE OUT OF RANGE/)
  904 FORMAT(15X,23HTIME VALUE OUT OF ORDER/)
  905 FORMAT(15X,5HNOISE/)
  906 FORMAT(15X,53HEOS STATUS REGISTER HAS CHANGED DURING COPY OPERATIO
     +N/)
  907 FORMAT(15X,14HSIGNATURE LOSS/)
   97 FORMAT(10X,14HERROR DETECTED,2I10)
   98 FORMAT(10X,10HVCDB ARRAY/15X,17HVEHICLE SIGNATURE/
     +15X,30HTOTAL SIGNATURE TIME            ,I10,16H ( * 4 MILLISEC)/
     +15X,30HFIRST PEAK AMPLITUDE            ,I10/
     +15X,30HTIME FROM TURN ON TO 1ST PEAK ,I10,16H ( * 4 MILLISEC)/
     +15X,30HSECOND PEAK AMPLITUDE           ,I10/
     +15X,30HTIME FROM TURN ON TO 2ND PEAK ,I10,16H ( * 4 MILLISEC)/
     +15X,30HLAST PEAK AMPLITUDE             ,I10/
     +15X,30HTIME FROM TURN ON TO LAST PEAK,I10,16H ( * 4 MILLISEC)/15X,
     +19HT(C1)+T(C4)-T(C3) =,I10,3X,2HBO//12X,62HTHE FOLLOWING ARE DETER
     +MINED IN THE STOPPED VEHICLE CLASSIFIER/15X,F8.5,5H*A1 =,I10,3X,
```

```
     +3HB13/15X,F8.5,5H*A2 =,I10,3X,3HB13/15X,F8.5,5H*A1 =,I10,3X,3HB13/
     +15X,F8.5,5H*A2 =,I10,3X,3HB13/
     +24X,4HH1 =,I10,3X,3HB13/24X,4HH2 =,I10,3X,3HB13//12X,
     +61HTHE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER/
     +15X,F8.5,5H*A2 =,I21,3X,3HB13/15X,F8.5,5H*A1 =,I21,3X,3HB13/
     +25X,3HH =,I21,3X,3HB13/15X,F8.6,16H*(T(C2)-T(C1)) =,I10,3X,3HB13/
     +15X,F8.5,16H*(T(C3)-T(C2)) =,I10,3X,3HB13/25X,3HF =,I21,3X,3HB13/
     +24X,4HK4 =,I21,3X,3HB13/24X,4HK5 =,I21,3X,3HB13///)
  99 FORMAT(1H0,10X,14HERROR DETECTED,2I10)
     END
```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
CON	R	ARRAY	00001 V	33	ERE	R	SCALR	00000 V	1	ERE		SPROG	00000 P	
I	I	SCALR	*00025 V	DUMMY	I1	I	SCALR	00023 V	1	I2	I	SCALR	00024 V	1
J	I	SCALR	00022 V	1	J1	I	SCALR	*00026 V	DUMMY	K	I	ARRAY	*00027 V	DUMMY
PERR	I	ARRAY	00001 C	17	SDB	I	ARRAY	00044 C	256	VCCO	I	ARRAY	0003A C	10
VCDB	I	ARRAY	00012 C	25	VCDT	I	SCALR	00144 C	1	VCLBG	I	ARRAY	0002B C	15
VCOUT	I	SCALR	00000 C	1										

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
20	0000A	25	0008D	30	00093	97	000FB	98	000F3	99	001F0
900	000A0	901	000A6	902	000AF	903	000BA	904	000C4	905	000CE
906	00003	907	000E4								

LOCAL VARIABLES (37 WORDS):

00000 ERE 00001 CON 00022 J 00023 I1 00024 I2

BLANK COMMON (325 WORDS):

00000 VCOUT 00001 PERR 00012 VCDB 0002B VCLBG 0003A VCCO 00044 SDB
00144 VCDT

ENTRY POINTS:

00000 ERE

EXTERNAL SUBPROGRAMS REQUIRED:

9BCDWRIT 9ENDIO 9IODATA 9SETUPN

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

	DEC WORDS	HEX WORDS	
GENERATED CODE:	506	001FA	
CONSTANTS:	0	00000	
LOCAL VARIABLES:	37	00025	
TEMPS:	4	00004	
TOTAL PROGRAM:	547	00223	(PLUS BLANK COMMON)

```
C
      SUBROUTINE VCLAS
C*********************************************************************
C  THIS PROGRAM CLASSIFYS VEHICLE SIGNATURES AS BUS OR NON-BUS. KEY
C DATUM ITEMS USED IN THIS EVALUATION MAY BE OUTPUTTED DEPENDING ON THE
C PRINTOUT CONTROL PARAMETER.
C*********************************************************************
C******  SUBROUTINES CALLED BY ROUTINE VCLAS
C FRE-RECORDS ERRORS DETECTED IN VEHICLE CLASSIFICATION ROUTINE
C*********************************************************************
C**********************  VARIABLES  ******************************
C CON-FLOATING POINT CONSTANTS
C FO= T(C1)*S/T(C1)+T(C4)-T(C3) WHERE S IS THE SCALE FACTOR
C F1F2-F1 + F2
C H1T1-EVALUATION TERM OF H1 FUNCTION
C H1T2-EVALUATION TERM OF H1 FUNCTION
C H2T1-EVALUATION TERM OF H2 FUNCTION
C H2T2-EVALUATION TERM OF H2 FUNCTION
C SDB-SIGNATURE DATA BASE
C VC31-G1 COEFFICIENT FOR H1 FUNCTION
C VC32-G2 COEFFICIENT FOR H1 FUNCTION
C VC41-G1 COEFFICIENT FOR H2 FUNCTION
C VC42-G2 COEFFICIENT FOR H2 FUNCTION
C VCCO-HOLDS COEFFICIENTS FOR THE EVALUATION OF THE K4 AND K5 FUNCTIONS
C VCST1-EVALUATION TERM FOR F1F2
C VCST2-EVALUATION TERM FOR FO
C VCST3-EVALUATION TERM FOR FO
C VCDR-VEHICLE SIGNATURE DATA BASE
C VCDT-VEHICLE DETECTOR INDEX NUMBER
C VCHD-HIGHEST DETECTOR INDEX NUMBER EXPECTED
C VCLD-LOWEST DETECTOR INDEX NUMBER EXPECTED - 1
C VCLBG-VEHICLE CLASSIFIER LOG
C VCVC-VEHICLE COUNTER
C VEH-VEHICLE STATUS    VEH=1 VEHICLE SIGNATURE BELONGS TO NON-BUS
C                       VEH=0 VEHICLE SIGNATURE BELONGS TO BUS
C*********************************************************************
      COMMON VCOUT,PERR(17)
      COMMON VCDR(25),VCLBG(15),VCCO(10),SDB(32,8),VCDT
      INTEGER VCOUT,PERR
      INTEGER VCOB,VCDT,VCHD,VCLD,VCLBG,VCCO,VCVC,SDB,VCST1,VCST2,VCST3,
     +VC31,VC32,VC41,VC42,H1T1,H1T2,H2T2,H2T1
      INTEGER VCC1,VCC2,VCC3,VCC4,VCC5,VEH,FO,F1F2
      DATA VCHD,VCLD/4,0/
      DIMENSION VCVC(32),CON(33)
      DATA (CON(J),J=1,33)/2.5,0.43,.7,0,0,0,0,0,2.73,0.876,2.52,0.139,
     +0,0,0,0,0,0,0.277,0.0198,0.228,0.947,0.96,1.66,1.34,2.17,4.1,
     +1.444,3.93,6.47,3.0,5.0/
C
      IF(VCOUT) 3,2,1
   01 WRITE(9,904)
      GO TO 3
   02 WRITE(9,93)
C            CLEAR VCDR ARRAY
   03        DO 5 I = 1,25
             VCDR(I) = 0
   05        CONTINUE
C
```

```
C       INITIAL VARIABLES
        VCST1 = CON(1)*16. + 0.5
        VCST2 = CON(2)*16. + 0.5
        VCST3 = CON(3)*16. + 0.5
        VC31 = CON(9) * 4096
        VC32 = CON(10) * 4096
        VC41 = CON(11) * 4096
        VC42 = CON(12) * 4096
        DO 7 L = 1,10
        VCCO(L) = CON(L+19) * 4096
   07   CONTINUE
        H1T1 = CON(30) * 4096
        H1T2 = CON(31) * 4096
        H2T1 = CON(32) * 4096
        H2T2 = CON(33) * 4096
        F0 = 0
        F1F2 = 0
        CON(15) = CON(23)
        CON(16) = CON(22)
        CON(18) = CON(20)
        CON(19) = CON(21)
C
C
C
C       IS THE SPECIFIED DETECTOR IN RANGE
        IF(VCDT.GT.VCHD) GO TO 10
        IF(VCDT.GT.VCLD) GO TO 20
C               VEHICLE CLASSIFIER ERROR MODE 1
C               DETECTOR SPECIFIED DOES NOT EXIST
   10           CALL ERE(1,VCDT,PERR)
                GO TO 90
C
C       CHECK EOS STATUS REGISTER
   20   IF (SDB(VCDT,8).NE.0) GO TO 28
        WRITE(9,91)
        GO TO 90
C
C       COPY VEHICLE SIGNATURE DATA BASE
   28           DO 30 I = 1,8,1
                VCDB(I) = SDB(VCDT,I)
   30           CONTINUE
C
C       OPTIONAL PRINTOUT
        IF (VCOUT) 32,32,31
   31   WRITE(9,92)(VCDB(J),J=1,7)
   32   CONTINUE
C
C       RECHECK EOS STATUS REGISTER
        IF(SDB(VCDT,8).NE.0) GO TO 40
                CALL ERE(6,VCDT,PERR)
C               CHECK THIRD PEAK AMPLITUDE
                IF (SDB(VCDT,6).NE.0) GO TO 40
                CALL ERE(7,VCDT,PERR)
                GO TO 90
C
C       RESET EOS STATUS REGISTER
   40   SDB(VCDT,8) = 0
C
```

```
C     TEST FOR NOISE
      IF (VCDB(1).GT.32) GO TO 50
C
C             TOTAL TIME OF SIGNATURE TOO SHORT FOR VEHICLE
              CALL ERE(5,VCDT,PERR)
              VCLOG(1) = VCLOG(1) + 1
C             NOTE VCLOG IS A DOUBLE PRECISION ARRAY ON THE IMP
              GO TO 90
C
C     INCREMENT VEHICLE COUNTER
   50 VCVC(VCDT) = VCVC(VCDT) + 1
C
C     GROSS FEATURES TEST
      IF (VCDB(6).NE.0) GO TO 60
C             SIGNATURE FAILS GROSS FEATURES TEST
      IF (VCDUT) 53,52,52
   52 WRITE(9,900)
   53 VCLOG(2) = VCLOG(2) + 1
C             VCLOG IS A DOUBLE PRECISION ARRAY ON THE IMP
              GO TO 90
C
C     CHECK MAGNITUDE OF SIGNATURE DATA
C             CHECK MAGNITUDE OF AMPLITUDE DATA
   60         DO 61 I = 2,6,2
              IF (VCDB(I).LT.256) GO TO 61
              CALL ERE(2,VCDT,PERR)
              GO TO 90
   61         CONTINUE
C             CHECK MAGNITUDE OF TIME DATA
              DO 62 I = 1,7,2
              IF (VCDB(I).LT.4096)  GO TO 62
              CALL ERE(3,VCDT,PERR)
              GO TO 90
   62         CONTINUE
C
C     CHECK FOR TIME OVERFLOW
   70 IF (VCDB(1).EQ.4096) GO TO 200
              GO TO 100
C
C
C     RETURN TO PMON   VEHICLE CLASSIFIED AS NON-BUS
   90 VEH = 1
      RETURN
C
C
C
C     MOVING/STOPPED TEST
C
C
C
C     INSURE TIME DATA WORDS IN PROPER ORDER
  100 IF(VCDUT) 104,103,103
  103 WRITE(9,94)
  104 IF (VCDB(5).LT.VCDB(3)) GO TO 105
      IF (VCDB(7).LT.VCDB(5)) GO TO 105
      IF (VCDB(1).LT.VCDB(7)) GO TO 105
      GO TO 110
  105         CALL ERE(4,VCDT,PERR)
```

```
                VEH = 1
                RETURN
C       FORM T(C1) + T(C4) = T(C3)
 110    VCDB(8) = VCDB(3) + VCDB(1) = VCDB(7)
C       FORM F0
        F0 = (VCDB(3)*16) / VCDB(8)
C       CHECK FOR OVERFLOW
        IF (F0.LE.32767) GO TO 120
C               OVERFLOW DETECTED
C               USE STOPPED VEHICLE CLASSIFIER
                CALL ERE(0,VCDT,PERR)
                GO TO 200
C       VEHICLE MAY BE MOVING IF 0.43 IS LESS THAN F0 IS LESS THAN 0.7
 120    IF(F0.GE.VCST3) GO TO 200
        IF (F0.LT.VCST2) GO TO 200
C       FORM F1 + F2
        F1F2 = ((VCDB(7)-VCDB(3))*16) / VCDB(8)
C       CHECK FOR OVERFLOW
        IF (F1F2.LE.32767) GO TO 130
C               OVERFLOW DETECTED
C               USE STOPPED VEHICLE DETECTOR
                CALL ERE(0,VCDT,PERR)
                GO TO 200
C       VEHICLE IS CONSIDERED MOVING IF F1F2 IS LESS THAN OR EQUAL TO 2.5
 130    IF (F1F2.GT.VCST1) GO TO 200
        GO TO 300
C
C
C
C       STOPPED VEHICLE CLASSIFIER
C
C
C       INCREMENT STOPPED VEHICLE COUNTER
 200    IF(VCOUT) 202,201,201
 201    WRITE(9,95) F0,F1F2
 202    VCLOG(3) = VCLOG(3) + 1
C       NOTE VCLOG IS A DOUBLE PRECISION ARRAY ON THE IMP
C
C       FORM VC31 * A1
        VCDB(9) = VC31 * VCDB(4)
C       FORM VC32 * A2
        VCDB(10) = VC32 * VCDB(6)
C       FORM ((VC31*A1)+(VC32*A2)) / A0
        VCDB(13) = (VCDB(9)+VCDB(10)) / VCDB(2)
C       CHECK FOR OVERFLOW
        IF (VCDB(13).LE.32767) GO TO 210
C               OVERFLOW DETECTED
C               VEHICLE NON-BUS
                CALL ERE(0,VCDT,PERR)
                GO TO 410
C
C       VEHICLE MAY BE A BUS IF VCDB(13) LIES BETWEEN H1T1 AND H1T2
 210    IF (VCDB(13).LT.H1T1) GO TO 410
        IF (VCDB(13).GE.H1T2) GO TO 410
C       FORM VC41 * A1
        VCDB(11) = VC41*VCDB(4)
C       FORM VC42 * A2
        VCDB(12) = VC42*VCDB(6)
C       FORM ((VC41*A1) + (VC42*A2))/A0
        VCDB(14) = (VCDB(11) + VCDB(12))/VCDB(2)
C       CHECK [illegible] OVERFLOW
```

```
      IF (VCDB(14).LE.32767) GO TO 220
C           OVERFLO- DETECTED
C           VEHICLE NON-BUS
            CALL ERE(0,VCDT,PERS)
            GO TO 410
C
C     VEHICLE IS BUS IF VCDB(14) LIES BETWEEN H2T1 AND H2T2
  220 IF (VCDB(14).LE.H2T1) GOTO 410
      IF (VCDB(14).GT.H2T2) GO TO 410
      GO TO 501
C
C
C
C     MOVING VEHICLE CLASSIFIER
C
C
  300 IF(VCOUT) 302,301,301
  301 WRITE(9,96) F0,F1F2
  302 VCC1 = 1
      VCC2 = 2
      VCC3 = 3
      VCC4 = 4
      VCDB(22) = 0
      VCLOG(4) = VCLOG(4) + 1
C     VCLOG IS A DOUBLE PRECISION ARRAY ON THE TMP
C     VC<
C           FORM C4 = A2
  310 VCDB(15) = VCCO(VCC4)*VCDB(6)
C           FORM C3 = A1
      VCDB(16) = VCCO(VCC3)*VCDB(4)
      VCDB(17) = (VCDB(16)-VCDB(15))/VCDB(2)
C     CHECK FOR OVERFLOW
      IF (VCDB(17).GT.32767) GO TO 600
C           FORM C1*(T(C2) - T(C1))
      VCDB(18) = VCCO(VCC1)*(VCDB(5)-VCDB(3))
C           FORM C2 * (T(C3) - T(C2))
      VCDB(19) = VCCO(VCC2)*(VCDB(7)-VCDB(5))
      VCDB(20) = (VCDB(19) - VCDB(18))/VCDB(8)
C     CHECK FOR OVERFLOW
      IF (VCDB(20).GT.32767) GO TO 600
C
C           FORM SUM
C     VEHICLE IS NON-BUS IF RESULTING SUM IS GREATER OR EQUAL TO ZERO
C
      IF(VCC1.EQ.6) GO TO 320
      VCDB(21) = VCDB(20) + VCDB(17) + VCCO(5)
      IF(VCDB(21).GE.0) GO TO 400
      GO TO 349
  320 VCDB(22) = VCDB(20) + VCDB(17) + VCCO(10)
      IF(VCDB(22).GE.0) GO TO 400
  349 IF(VCOUT) 351,351,350
  350 WRITE(9,302) VCDB(8),((CON(J),VCDB(J)),J=15,16),VCDB(17),(CON(J),
     +VCDB(J)),J=18,19),(VCDB(J),J=20,22)
  351 IF(VCC1.EQ.6) GO TO 500
      CON(15) = CON(28)
      CON(16) = CON(27)
      CON(18) = CON(25)
      CON(19) = CON(26)
      VCC1 = 6
      VCC2 = 7
      VCC3 = 8
      VCC4 = 9
      VCDB(21) = 0
```

```
      GO TO 310
C
C
 500  IF(VCOUT) 510,505,505
C     VEHICLE IS NON-BUS
 400  IF(VCOUT) 420,402,401
 401  WRITE(9,902) VCDB(8),((CON(J),VCDB(J)),J=15,16),VCDB(17),((CON(J),
     +VCDB(J)),J=18,19),(VCDB(J),J=20,22)
 402  WRITE(9,97)
      GO TO 420
 410  IF(VCOUT) 420,412,411
 411  WRITE(9,98) VCDB(8),((CON(J),VCDB(J)),J=9,12),(VCDB(J),J=13,14)
 412  WRITE(9,97)
 420  VCLOG(5) = VCLOG(5) + 1
C     NOTE VCLOG IS A DOUBLE PRECISION ARRAY ON THE IMP
      VEH = 1
      RETURN
C
C     VEHICLE IS A BUS
 501  IF(VCOUT) 510,505,504
 504  WRITE(9,98) VCDB(8),((CON(J),VCDB(J)),J=9,12),(VCDB(J),J=13,14)
 505  WRITE(9,99)
 510  VEH = 0
      VCLOG(6) = VCLOG(6) + 1
      RETURN
C
C     OVERFLOW ERROR
 600  VEH = 1
      CALL ERE(0,VCDT,PERR)
      RETURN
C
C
 904  FORMAT(1H1)
 900  FORMAT(10X,18HGROSS TEST FAILURE///)
  91  FORMAT(1H0,24HEOS STATUS REGISTER ZERO//33HVEHICLE CLASSIFICATION
     +TERMINATED)
  92  FORMAT(1H0,5X,18HVEHICLE-CLASSIFIER//15X,17HVEHICLE SIGNATURE/
     +15X,30HTOTAL SIGNATURE TIME           ,I10,16H ( * 4 MILLISEC)/
     +15X,30HFIRST PEAK AMPLITUDE           ,I10/
     +15X,30HTIME FROM TURN ON TO 1ST PEAK ,I10,16H ( * 4 MILLISEC)/
     +15X,30HSECOND PEAK AMPLITUDE          ,I10/
     +15X,30HTIME FROM TURN ON TO 2ND PEAK ,I10,16H ( * 4 MILLISEC)/
     +15X,30HLAST PEAK AMPLITUDE            ,I10/
     +15X,30HTIME FROM TURN ON TO LAST PEAK,I10,16H ( * 4 MILLISEC)/)
  93  FORMAT(1H0,5X,18HVEHICLE-CLASSIFIER)
  94  FORMAT(10X,19HMOVING/STOPPED TEST)
  95  FORMAT(10X,15HSTOPPED VEHICLE,10X,3HF0=,I10,4H  BO,6X,6HF1*F2=,
     +I10,4H  BO//)
  96  FORMAT(10X,14HMOVING VEHICLE,10X,3HF0=,I10,4H  BO,6X,6HF1*F2=,I10,
     +4H  BO//)
  97  FORMAT(1H ,5X,7HNON-BUS//)
  98  FORMAT(15X,
     +19HT(C1)+T(C4)-T(C3) =,I10,3X,2HBO//15X,62HTHE FOLLOWING ARE DETER
     +MINED IN THE STOPPED VEHICLE CLASSIFIER/15X,F8.5,5H*A1 =,I10,3X,
     +3HB13/15X,F8.5,5H*A2 =,I10,3X,3HB13/15X,F8.5,5H*A1 =,I10,3X,3HB13/
     +15X,F8.5,5H*A2 =,I10,3X,3HB13/
     +24X,4HH1 =,I10,3X,3HB13/24X,4HH2 =,I10,3X,3HB13)
 902  FORMAT(15X,19HT(C1)+T(C4)-T(C3) =,I10,3X,2HBO//15X,
```

```
     +61HTHE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER/
     +15X,F8.5,5H*A2 =,I21,3X,3HB13/15X,F8.5,5H*A1 =,I21,3X,3HB13/
     +25X,3HG =,I21,3X,3HB13/15X,F8.5,16H*(T(C2)-T(C1)) =,I10,3X,3HB13/
     +15X,F8.5,16H*(T(C3)-T(C2)) =,I10,3X,3HB13/25X,3HF =,I21,3X,3HB13/
     +24X,4H<4 =,I21,3X,3HB13/24X,4H<5 =,I21,3X,3HB13///)
 99  FORMAT(1H ,5X,3HBUS//)
     END
```

NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS	NAME	TYPE	CLASS	HEX LOC	DEC WORDS
CON	R	ARRAY	00035 V	33	ERE		SPROG	EXTERN		FO	I	SCALR	00033 V	1
F1F2	I	SCALR	00034 V	1	H1T1	I	SCALR	0002A V	1	H1T2	I	SCALR	00028 V	1
H2T1	I	SCALR	00020 V	1	H2T2	I	SCALR	0002C V	1	I	I	SCALR	00057 V	1
J	I	SCALR	00056 V	1	L	I	SCALR	00058 V	1	PERR	I	ARRAY	00001 C	17
SDB	I	ARRAY	00044 C	256	VCC0	I	ARRAY	0003A C	10	VCC1	I	SCALR	0002E V	1
VCC2	I	SCALR	0002F V	1	VCC3	I	SCALR	00030 V	1	VCC4	I	SCALR	00031 V	1
VCC5	I	UNUSED			VCDB	I	ARRAY	00012 C	25	VCDT	I	SCALR	00144 C	1
VCHD	I	SCALR	00001 V	1	VCLAS		SPROG	00000 P		VCLAS	R	SCALR	00000 V	1
VCLD	I	SCALR	00002 V	1	VCLBG	I	ARRAY	0002B C	15	VCBUT	I	SCALR	00000 C	1
VCST1	I	SCALR	00023 V	1	VCST2	I	SCALR	00024 V	1	VCST3	I	SCALR	00025 V	1
VCVC	I	ARRAY	00003 V	32	VC31	I	SCALR	00026 V	1	VC32	I	SCALR	00027 V	1
VC41	I	SCALR	00028 V	1	VC42	I	SCALR	00029 V	1	VEH	I	SCALR	00032 V	1

LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC	LABEL	HEX LOC
1	00005	2	0000A	3	0000E	5	00015	7	00040	10	00065
20	0006B	28	00073	30	0007D	31	00082	32	0008F	40	00A0
50	00AD	52	00B3	53	00B7	60	00B9	61	00C7	62	00D9
70	00DB	80	00DF	91	00F7	92	00FA	93	00F7	94	0100
95	0108	96	0031B	97	0032E	98	00334	99	003D4	100	00E2
103	00E4	104	00E8	105	00F2	110	00FA	120	0010B	130	0011F
200	00123	201	00125	202	0012B	210	0013F	220	00158	300	0015F
301	00161	302	00167	310	00172	320	0019C	340	001A2	350	001A4
351	001CD	400	001E6	401	001F9	402	00212	410	00217	411	0021A
412	00235	420	00239	500	001F3	501	0023D	504	00240	505	00258
510	0025F	600	00263	900	0026E	902	0037B	904	00268		

LOCAL VARIABLES (89 WORDS):

00000	VCLAS	00001	VCHD	00002	VCLD	00003	VCVC	00023	VCST1
00024	VCST2	00025	VCST3	00026	VC31	00027	VC32	00028	VC41
00029	VC42	0002A	H1T1	0002B	H1T2	0002C	H2T2	0002D	H2T1
0002E	VCC1	0002F	VCC2	00030	VCC3	00031	VCC4	00032	VEH
00033	FO	00034	F1F2	00035	CON	00056	J	00057	I
00058	L								

BLANK COMMON (325 WORDS):

00000	VCBUT	00001	PERR	00012	VCDB	0002B	VCLBG	0003A	VCC0
00044	SDB	00144	VCDT						

ENTRY POINTS:

00000 VCLAS

EXTERNAL SUBPROGRAMS REQUIRED:

ERE 9BCDARIT 9ENDIAL 9IBDATA 9RT81 9SETUP0

```
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

                    DEC       HEX
                   WORDS     WORDS
                   -----     -----
  GENERATED CODE:   906     0030A
       CONSTANTS:    11      0000B
 LOCAL VARIABLES:    89      00059
           TEMPS:     6      00006
                   -----     -----
   TOTAL PROGRAM:  1092      00444   (PLUS BLANK COMMON)
```

```
LBAD (EF,(MAIN),(9ERR,F4LIB)),(MAP);
, (UNSAT,(F4LIB)),(LMN,PADVC),(PERM)
```

PBOVC

```
CSEC    32AE 0
CSEC    20AA 0
DSEC    2000 0   F4:COM
UDEF    32AE 0   4MAIN
CSEC    33C4 0
CSEC    20AE 0
DSEC    2000 0   F4:COM
 DEF    33C4 0   ERF
CSEC    35BE 0
CSEC    20D8 0
DSEC    2000 0   F4:COM
 DEF    35BE 0   VCLAS
UDEF    2E38 0   9ERR9R
CSEC    2E35 0
LDEF    39E4 0   9INITIAL
LDEF    3A15 0   9BCDREAD
LDEF    406D 0   9I8DATA
LDEF    4099 0   9ENDI8I
LDEF    3A4F 0   9BCDXRIT
LDEF    412E 0   9ST9P
LDEF    39D6 0   9SETUPN
LDEF    39BE 0   9SETUPC
LDEF    39B6 0   9RT8I
LDEF    416A 0   7ERRHEAD
LDEF    4194 0   7ERRTEXT
LDEF    41C4 0   7PAC
LDEF    41BE 0   7PRC
LDEF    41C2 0   7PRQ
LDEF    4134 0   7ST9P
LDEF    39A4 0   EXIT
LDEF    39AE 0   9IT9D
LDEF    39AE 0   9IT9R
LDEF    39B7 0   9DT8I
LDEF    39CD 0   9SETUP1
LDEF    39C8 0   9SETUP2
LDEF    39D7 0   7SET
LDEF    39D6 0   9SETUP4
LDEF    2F03 0   8ABORTEX
LDEF    2F04 0   8ABRTSEV
LDEF    2F05 0   8E9FEXIT
LDEF    2F06 0   8ERREXIT
LDEF    2F07 0   8FLOVTRG
LDEF    2F08 0   8I6TRIG
LDEF    2F09 0   8SENLITE
LDEF    2FFC 0   8TINIT
LDEF    2FCE 0   6ENDEQ
LDEF    2FCF 0   6ERREQ
LDEF    40DC 0   7EHFABRT
LDEF    40FE 0   7UNITADR
LDEF    2F9D 0   8BCDBUF
LDEF    2FC5 0   8DCBADR
LDEF    2F1A 0   8I8DLINK
LDEF    2F17 0   8TALPHA
LDEF    3A9D 0   9IEDIT
LDEF    3A2A 0   7READ
LDEF    3A18 0   9BCDRDFE
LDEF    3A14 0   9READ
LDEF    3A92 0   9BEDIT
LDEF    3A4E 0   9PRINT
```

LDEF	3AFE	9PRINT
LDEF	41A3	7FRFCALK
LDEF	2FC4	[illegible]
LDEF	2FC0	8ENDIO
LDEF	2FC9	8INPTERM
LDEF	2FC1	8IODADDR
LDEF	2FC2	8IODTYPE
LDEF	2FC3	8IOENLOC
LDEF	2F3E	8TEDIT
LDEF	403C	9DATA
LDEF	2FC8	6NOELMTS
LDEF	406E	7IODATUM
LDEF	2F1C	8TOETA
LDEF	4018	9IOLUSA
LDEF	418F	7FRPOR
LDEF	415C	7GETMODE
LDEF	2FCA	6DATLINK
LDEF	40CD	9REWIND
LDEF	40CE	9ENDFILE
LDEF	2F22	8MSGBUF
LDEF	2FC6	8UNITNAM
LDEF	2FCC	6DCBNAME
LDEF	414E	7BINDEC
LDEF	2FC8	8UNITVAL
LDEF	2F36	8T0
LDEF	2FCA	8TERROR
LDEF	41E6	7BUFOUT
LDEF	41E4	7BUFOUTC
LDEF	41E1	7ERINIT
LDEF	41CC	7PHC
LDEF	41DD	7PHL
LDEF	2FE9	8T1
LDEF	2FEA	8T2
LDEF	2FEB	8T3
LDEF	2FEC	8T4
LDEF	2FED	8T5
LDEF	2FEE	8T6
LDEF	2FEF	8T7
LDEF	2FF0	8T8
LDEF	2FF1	8T9
LDEF	2FF2	8T10
LDEF	2FF3	8T11
LDEF	2FF4	8T12
LDEF	2FF5	8T13
LDEF	2FF6	8T14
LDEF	2FF7	8T15
LDEF	2FF8	8T16
LDEF	2FF9	8T17
LDEF	2F3E	6BINBUF
LDEF	2FD1	6EGFCALL
LDEF	2FD2	6EGFUADR
LDEF	2FD3	6EGFUADR
LDEF	2FD4	6EGFUTRG
LDEF	2FD0	6RFCSIZE
LDEF	2FD5	8TDUMP
LDEF	2FDA	8TINPUT
LDEF	2FEE	6DISCBUF
	2D46	SEGLO0
	3047	SEGHI0
	3D0D	SEGLO1
	402F	SEGHI1
	4400	SEGLO2
	42FF	SEGHI2

PROGRAM LISTING OF PBD33J - BUS PREEMPT CONTROL UNIT
PROGRAM FOR MINNEAPOLIS INSTALLATION

APPENDIX D. FIGURE 8.8

```
XXXX
```

```
007    $     BJE                                                    /10. . .702.........
0078*  $     OBJECT                                                 Y10.780070275ACNTL000
0080*  $     OBJECT                                                 Y10.781070275LIST0000
0082*  $     OBJECT                                                 Y10.781070275REGINT00
0084*  $     OBJECT                                                 Y10.782070275INST0000
0086*  $     OBJECT                                                 Y10.78207027500000000
0089*  $     OBJECT                                                 Y10.782070275SYMVAL00
0090*  $     OBJECT                                                 Y10.783070275EXTSYM00
0092*  $     OBJECT                                                 Y10.783070275NXTSYM00
0094*  $     OBJECT                                                 Y10.783070275ADDSYM00
0096*  $     OBJECT                                                 Y10.784070275ADPART00
0098*  $     OBJECT                                                 Y10.784070275ERREPT00
0100*  $     OBJECT                                                 Y10.784070275DIRECT00
0102*  $     OBJECT                                                 Y10.785070275BOCNTL00
0104*  $     OBJECT                                                 Y10.785070275GENRLM00
0106   $     EXECUTE
0107   $     SET      6,19
0108   $     LIMITS   15,23K,,7500
0109   $     REMOTE   06
0110   $     FILE     10,X1S,40L
0111   $$    PRMFL    09,R/W,S,D00391/AOUT
0112   $$    SELECT   D00391/EAC0826$######
0113*  $     INCODE   IBMEL
0114   $     INCODE   IBMEL
0115   $$    SELECT   D00391/EBP0826$######
0116*  $     INCODE   IBMEL
0117   $     INCODE   IBMEL
0118   $$    SELECT   D00391/EER1001$######
0119*  $     INCODE   IBMEL
0120   $     INCODE   IBMEL
0121   $$    SELECT   D00391/EES0826$######
0122*  $     INCODE   IBMEL
0123   $     INCODE   IBMEL
0124   $$    SELECT   D00391/EPP0819$######
0125*  $     INCODE   IBMEL
0126   $     INCODE   IBMEL
0127   $$    SELECT   D00391/ETP0826$######
0128*  $     INCODE   IBMEL
0129   $     INCODE   IBMEL
0130   $$    SELECT   D00391/EPR0826$######
0131*  $     INCODE   IBMEL
0132   $$    SELECT   D00391/EVC1001$######
0133*  $     INCODE   IBMEL
0134   $     INCODE   IBMEL
0135   $$    SELECT   D00391/EEB0826$######
0136*  $     INCODE   IBMEL
0137   $     INCODE   IBMEL
0138   $$    SELECT   D00391/EBX0826$######
0139*  $     INCODE   IBMEL
0140   $     INCODE   IBMEL
0141   $$    SELECT   D00391/EDB1010$######
0142*  $     INCODE   IBMEL
0143   $     IF       /19,ENDJOB
0144   $     OPTION   FORTRAN,NOMAP
0145   $     LIBRARY  L1
0146   $$    SELECT   G02771/NATIONAL/XREF
0147*  $     OBJECT                                                 Y16.706071475......00
0149   $     EXECUTE
0150   $     LIMIT    ,20K
0151   $$    PRMFL    L1,R,R,USERLIB/UTILITY/BUTIL
0152   $     FILE     10,X1R
0153   $     FILE     11,X2R,40L
0154   $     FILE     S1,X3R,44R
0155   $     ENDJOB
  TOTAL CARD COUNT THIS JOB = 005812
```

```
TOTAL CARD COUNT THIS JOB = 005912

* BEGIN ACTIVITY -01-  GELOAD   10/10/75   SW=010000000000
 * NORMAL TERMINATION     AT 011743 I=5020 SW=010000000000

     START 15.953      LINES     21      PROC  0.0001       I/O   0.000    IU  5   MEMORY    5K
     STOP  15.954      LIMIT   5000      LIMIT 0.0100       LIMIT          CU  5   M*T      44
     SWAP   0.000
     LAPSE  0.001   FC D TYPE     BUSY    IP/AT    FP/RT    IS/#C MS/#E    ADDRESS T#

                    R* R D191 *     12        0        0        1     1    0-08-03
                    P*  SYOUT
                    L* R D191 *    338        0        0      528  528R    0-08-05

      LIST     21 LINES AT STA.  G5

RESOURCE UNITS USED, ACTIVITY -01-     0.00
***RESOURCE UNITS USED, JOB TOTAL-     0.00

* BEGIN ACTIVITY -02-  GELOAD   10/10/75   SW=014000200000
 * NORMAL TERMINATION     AT 013613 I=5020 SW=014000200000

     START 16.009      LINES   6152      PROC  0.0731       I/O   0.010    IU  5   MEMORY   23K
     STOP  16.531      LIMIT   7500      LIMIT 0.1500       LIMIT          CU  5   M*T    41187
     SWAP   0.066
     LAPSE  0.523   FC D TYPE     BUSY    IP/AT    FP/RT    IS/#C MS/#E    ADDRESS T#

                    I* R D191 *  18773        0      252      252   252    0-08-03
                    R* R D191 *    917        0        0       46    46    0-08-03
                    06  SYOUT
                    10 S D191 *   2963        0        0      480   480    0-08-04
                    09 R D191 P    861        0       18       20    20    0-08-05
                    P*  SYOUT
                    L* R D191 *   1184        0        0      528   528R   0-08-05

      LIST     22 LINES AT STA.  G5
      RC-06  6130 LINES AT STA.  G5

RESOURCE UNITS USED, ACTIVITY -02-     2.82
***RESOURCE UNITS USED, JOB TOTAL-     2.82

* BEGIN ACTIVITY -03-  GELOAD   10/10/75   SW=000000200000
SORT ENGAGED - VERSION: F/8
            FILE 11 UNSUITABLE FOR COLLATION.
MEMORY: 023416  LINKS: 00044 INPUT: 0320/DOUBLE  OUTPUT: 0320/DOUBLE  COLLATION: 0362/DOUBLE  TOURNAMENT: 000541 ENTRIES
            END OF FILE ON 10.  RECORDS READ = 00006871.  RECORDS ACCEPTED = 00006871.  RECORDS DELETED = 00000000.
            MERGING 000007 STRINGS 08 WAYS.
            LINK UTILIZATION.      ALLOCATED 00044  BORROWED 00000  USED 00008
            RECORDS INPUT = 00006871.  RECORDS OUTPUT = 00006871.
SORT TERMINATES.
 * NORMAL TERMINATION     AT 013403 I=5020 SW=000000200000

     START 16.533      LINES   2113      PROC  0.0050       I/O   0.004    IU  5   MEMORY   20K
     STOP  16.547      LIMIT   5000      LIMIT 0.0500       LIMIT          CU  5   M*T    1126
     SWAP   0.000
     LAPSE  0.014   FC D TYPE     BUSY    IP/AT    FP/RT    IS/#C MS/#E    ADDRESS T#

                    10 R D191 *   1489        0        0      480   480    0-08-04
                    P* R D191 *     63        0        0        3     3    0-08-03
                    L1 R D191 P     66        0        0      120  120R    0-08-03
                    11 R D191 *   3525        0       87      480   480    0-08-02
```

```
P* SYOUT
R [illegible] *    71        0        [illegible]    [illegible]    0.[illegible]05

LIST      26 LINES AT STA.  G5
RC-06   2087 LINES AT STA.  G5

RESOURCE UNITS USED, ACTIVITY -03-      0.28
***RESOURCE UNITS USED, JOB TOTAL-      3.10
```

SNUMB = PB33J, ACTIVITY # = 01, REPORT CODE = 74, RECORD COUNT = 000021

```
ORIGIN   DATE MODULE ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                     SUBPROGRAMS INCLUDED IN DECK.

                $      OPTION  FORTRAN,NOMAP
                $      OPTIONS FORTRAN,NOMAP

                     SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

                                   RANGE              SIZE
                ALLOCATED CORE   000000 THRU 011777   012000
                   RELOCATABLE   007062 THRU 011777   002716
                $      SYSOUT  P*,ORG

                FCB AND BUFFER SPACE
                   AVAILABLE     000101 THRU 007061   006761
                   FILE CTRL BLKS 006732 THRU 007062  000131
                   MAXIMUM BUFFER SPACE REQUIRED      001202

                     3K. IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN   730517 F/8
                000204 LOCATIONS REQUIRED FOR LOAD TABLE
                EXECUTION PROGRAM ENTERED AT  011772   THROUGH  .FSETU
```

SNUMB = PB33J, ACTIVITY # = 02, REPORT CODE = 74, RECORD COUNT = 000022

```
ORIGIN   DATE MODULE ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION

                    SUBPROGRAMS INCLUDED IN DECK.

              $     OPTIONS FORTRAN,NOMAP

                    SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

                                      RANGE              SIZE
              ALLOCATED CORE    000000 THRU 055777      056000
                 RELOCATABLE    004422 THRU 055777      051356
              $      REMOTE  06
              $      FILE    10,X1S,40L
              $      PRMFL   09,R/W,S,D00391/AOUT

              FCB AND BUFFER SPACE
                 AVAILABLE      000101 THRU 004421      004321
                 FILE CTRL BLKS 004170 THRU 004422      000233
                 MAXIMUM BUFFER SPACE REQUIRED          003105

                 23K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN     730517 F/8
              001446 LOCATIONS REQUIRED FOR LOAD TABLE
              EXECUTION PROGRAM ENTERED AT  033506  THROUGH  .FSETU
```

SNUMB = PB33J, ACTIVITY # = 02, REPORT CODE = 06, RECORD COUNT = 006130

```
1 0000          .TITLE PBD33J,'          ... P B D 3 3 J ...'
```

```
 4 0000                    .LIST   LAC
 5 0000          ;
 6 0000          ;                    ... A S S E M B L Y   C O N S T A N T S ...
 7 0000          ;
 8 0000 0000 A AC0      =       0                ; ACCUMULATOR 0
 9 0000 0001 A AC1      =       1                ; ACCUMULATOR 1
10 0000 0002 A AC2      =       2                ; ACCUMULATOR 2
11 0000 0003 A AC3      =       3                ; ACCUMULATOR 3
12 0000         ;
13 0000 0001 A FC1      =       X'01             ; FUNCTION CODE 1, WRITE WORD 1
14 0000 0002 A FC2      =       X'02             ; FUNCTION CODE 2, WRITE WORD 2
15 0000 0003 A FC3      =       X'03             ; FUNCTION CODE 3, READ WORD 1
16 0000 0004 A FC4      =       X'04             ; FUNCTION CODE 4, READ WORD 2
17 0000         ;
18 0000 0010 A FC10     =       X'10             ; FUNCTION CODE 10, READ THUMBWHEEL 6
19 0000 0011 A FC11     =       X'11             ; FUNCTION CODE 11, READ THUMBWHEEL 1
20 0000 0012 A FC12     =       X'12             ; FUNCTION CODE 12, READ THUMBWHEEL 2
21 0000 0013 A FC13     =       X'13             ; FUNCTION CODE 13, READ THUMBWHEEL 3
22 0000 0014 A FC14     =       X'14             ; FUNCTION CODE 14, READ THUMBWHEEL 4
23 0000 0015 A FC15     =       X'15             ; FUNCTION CODE 15, READ THUMBWHEEL 5
24 0000         ;
25 0000         ;                                        NOTE: FUNCTION CODES ENDING IN
26 0000         ;                                              6 CANNOT BE USED FOR RIN
27 0000         ;                                              INSTRUCTIONS DUE TO
28 0000         ;                                              CONFLICT WITH TTY ROUTINE
29 0000         ;
30 0000 0017 A FC17     =       X'17             ; FUNCTION CODE 17, READ THUMBWHEEL 7
31 0000         ;
32 0000 0041 A FC41     =       X'41             ; FUNCTION CODE 41, RESET PULSE
33 0000 0042 A FC42     =       X'42             ; FUNCTION CODE 42, START PULSE
34 0000 0043 A FC43     =       X'43             ; FUNCTION CODE 43, STOP PULSE
35 0000 0044 A FC44     =       X'44             ; FUNCTION CODE 44, TURN ON LIGHT 1
36 0000 0045 A FC45     =       X'45             ; FUNCTION CODE 45, TURN OFF LIGHT 1
37 0000 0046 A FC46     =       X'46             ; FUNCTION CODE 46, TURN ON LIGHT 2
38 0000 0047 A FC47     =       X'47             ; FUNCTION CODE 47, TURN OFF LIGHT 2
39 0000         ;
40 0000         ;                    ... DISPLACEMENT CONSTANTS ...
41 0000 FFF4 A DM12     =       -12
42 0000 FFF9 A DM7      =       -7
43 0000 FFFC A DM4      =       -4
44 0000 FFFE A DM2      =       -2
45 0000 FFFF A DM1      =       -1
46 0000 0000 A D0       =       0
47 0000 0001 A D1       =       1
48 0000 0002 A D2       =       2
49 0000 0003 A D3       =       3
50 0000 0004 A D4       =       4
51 0000 0005 A D5       =       5
52 0000 0006 A D6       =       6
53 0000 0007 A D7       =       7
54 0000 0008 A D8       =       8
55 0000 0009 A D9       =       9
```

```
0000 000A A D10     =    10
0000 000B A D11     =    11
0000 000C A D12     =    12
0000 000D A D13     =    13
0000 000E A D14     =    14
0000 000F A D15     =    15
0000 0010 A D16     =    16
0000 0011 A D17     =    17
0000 0012 A D18     =    18
0000 0013 A D19     =    19
0000 0014 A D20     =    20
0000 0015 A D21     =    21
0000 0016 A D22     =    22
0000 0017 A D23     =    23
0000 0018 A D24     =    24
0000 0019 A D25     =    25
0000 001A A D26     =    26
0000 0074 A D116    =    116
0000 0076 A D118    =    118
0000        ;
0000 0004 A DETS    =    4          ; NUMBER OF DETECTORS FOR PBD33J .......
0000        ;
0000 0019 A DPMD    =    25         ; MINIMUM DURATION OF A PREEMPT
0000        ;                            ( 0.1 SEC UNITS)
0000        ;
0000        ;
0000        ;                  ... IMMEDIATE DATA CONSTANTS ...
0000 FF92 A CM110   =    -110
0000 FF9C A CM100   =    -100
0000 FFEC A CM20    =    -20
0000 FFF0 A CM16    =    -16
0000 FFF6 A CM10    =    -10
0000 FFFA A CM6     =    -6
0000 FFFE A CM2     =    -2
0000 FFFF A CM1     =    -1
0000 0000 A C0      =    0
0000 0001 A C1      =    1
0000 0002 A C2      =    2
0000 0003 A C3      =    3
0000 0004 A C4      =    4
0000 0005 A C5      =    5
0000 0006 A C6      =    6
0000 0007 A C7      =    7
0000 0008 A C8      =    8
0000 000F A C15     =    15
0000 0030 A C48     =    48
0000 007D A C125    =    125
```

```
0000                    .PAGE
0000          ;
0000          ;                    ... BOC CONDITION CODES ...
0000 0001 A CCEZ    =     X'1          ; CONDITION CODE 1 (AC0).EQ.0
0000 0002 A CCZP    =     X'2          ; CONDITION CODE 2 (AC0) ZERO, POSITIVE
0000 0003 A CCB0    =     X'3          ; CONDITION CODE 3 AC0 BIT 0 = 1
0000 0004 A CCB1    =     X'4          ; CONDITION CODE 4 AC0 BIT 1 ≠ 1
0000 0005 A CCNZ    =     X'5          ; CONDITION CODE 5 (AC0).NE.0
0000 0008 A CCSF    =     X'8          ; CONDITION CODE 8 STACK FULL
0000 000B A CCZN    =     X'B          ; CONDITION CODE 11 (AC0).LE.0
0000 000C A CCCU    =     X'C          ; CONDITION CODE 12 CCU IN SYSTEM
0000 000D A CCPH    =     X'D          ; CONDITION CODE 13 INITIALIZATION
0000                                   ;    PSEUDO HALT
0000 000E A CCCF    =     X'E          ; CONDITION CODE 14 CATCH ON THE FLY
0000          ;
0000          ;                    ... CONTROL FLAGS ...
0000 0001 A INEN    =     1            ; INTERRUPT ENABLE FLAG
0000 0002 A SELF    =     2            ; SELECT FLAG
0000          ;
0000          ;
```

```
 125 0000          .LIST   LBP
 126 0000      ;
 127 0000      ;                 ... B A S E   P A G E ...
 128 0000          .BSECT
 129 0000      ;                 ... TRANSFER TO RESTART INITIALIZATION ROUTINE...
 130 0000 2420 A       JMP     @AERSI
 131 0001      ;
 132 0001      ;                 ... TRANSFER TO GENERAL INTERRUPT SERVICING ROUTINE
 133 0001 2421 A       JMP     @AEGIS
 134 0002      ;
```

```
137 0002                .LIST   LBP
138 0002          ;
139 0002                .ASECT
140 0000 0020 A         .=X'20
141 0020 1000 A AERSI:  .WORD   ERSI        ; RESTART INITIALIZATION ROUTINE
142 0021 1041 A AEGIS:  .WORD   EGIS        ; GENERAL INTERRUPT SERVICING ROUTINE
143 0022 10C2 A AECCU:  .WORD   ECCU        ; CCU INITIALIZATION ROUTINE
144 0023 10E0 A AEMAD:  .WORD   EMAD        ; INITIAL RAM ADDRESS TEST ROUTINE
145 0024 1E00 A AEBKD:  .WORD   EBKD        ; BACKGROUND PROGRAM MONITOR
146 0025 10BF A AECER:  .WORD   ECER        ; CLOCK ERROR ROUTINE
147 0026 10F3 A AESCH:  .WORD   ESCH        ; EXECUTIVE SCHEDULING ROUTINE
148 0027 1200 A AEDIS:  .WORD   EDIS        ; DISCONNECT MONITOR
149 0028 1212 A AECOM:  .WORD   ECOM        ; COMMAND MONITORING ROUTINE
150 0029 123F A AESYN:  .WORD   ESYN        ; SYNCHRONIZATION ROUTINE
151 002A 1282 A AEMIC:  .WORD   EMIC        ; INTERCONNECT DATA MONITOR ROUTINE
152 002B 12A3 A AETIM:  .WORD   ETIM        ; EXECUTIVE TIMING ROUTINE
153 002C 131B A AEEOC:  .WORD   EEOC        ; END OF CYCLE ROUTINE
154 002D 12E0 A AESDW:  .WORD   ESDW        ; STATUS DISPLAY ROUTINE
155 002E 12B2 A AEDET:  .WORD   EDET        ; DETECTOR MONITORING ROUTINE
156 002F 110F A AEADC:  .WORD   EADC        ; AUTODISCONNECT ROUTINE
157 0030 1A00 A APMON:  .WORD   PMON        ; PREEMPT MONITOR ROUTINE
158 0031 1C19 A AVCLAS: .WORD   VCLAS       ; VEHICLE CLASSIFICATION ROUTINE
159 0032 1AE6 A APREQ:  .WORD   PREQ        ; PREEMPT REQUEST ROUTINE
160 0033 1A88 A APTER:  .WORD   PTER        ; PREEMPT TERMINATION ROUTINE
161 0034 1A4E A APCIM:  .WORD   PCIM        ; ROUTINE USED TO RESTORE CI
162 0035 1800 A ATPEX:  .WORD   TPEX        ; TIMING PLAN EXECUTION ROUTINE
163 0036 18C9 A ATCOPY: .WORD   TCOPY       ; TIMING PLAN COPY ROUTINE
164 0037 1866 A ATNCY:  .WORD   TNCY        ; NEXT TIMING PLAN CYCLE ROUTINE
165 0038 1896 A ATDIAL: .WORD   TDIAL       ; INIT ENTRY TO TNCY ROUTINE
166 0039 1F28 A AXRTC:  .WORD   XRTC        ; RTC DATA OUTPUT SEQUENCE ROUTINE
167 003A 1F32 A AXWDT:  .WORD   XWDT        ; RESET WATCHDOG TIMER ROUTINE
168 003B 1F36 A AXSDC:  .WORD   XSDC        ; FORMAT OUTPUT COMMANDS ROUTINE
169 003C 1F00 A ABRTC:  .WORD   BRTC        ; RTC DATA INPUT SEQUENCE ROUTINE
170 003D 1143 A ACOTF:  .WORD   COTF        ; CATCH ON THE FLY ROUTINE FOR CHECKOUT
171 003E          ;
172 003E          ;                 ... PROG. PANEL PROGRAM POINTERS ...
173 003E 1F0D A ABPAN:  .WORD   BPAN        ; PROG. PANEL PARAMETER INPUT ROUTINE
174 003F 1400 A AEPAN:  .WORD   EPAN        ; PROG. PANEL INITIALIZATION ROUTINE
175 0040 146E A AEPPM:  .WORD   EPPM        ; PROG PANEL MONITOR ROUTINE
176 0041 14C4 A AEPP1:  .WORD   EPP1        ; ROUTINE TO PROCESS PROG. PANEL PLAN 1
177 0042 1578 A AEPP2:  .WORD   EPP2        ; ROUTINE TO PROCESS PROG. PANEL PLAN 2
178 0043 162C A AEPP3:  .WORD   EPP3        ; ROUTINE TO PROCESS PROG. PANEL PLAN 3
179 0044 16E0 A AEPUP:  .WORD   EPUP        ; UPDATE PROG. PANEL DATA
180 0045 1733 A AEPBI2: .WORD   EPBI2       ; CONVERT 2 BINARY-CODED DIGITS
181 0046 1726 A AEPBI3: .WORD   EPBI3       ; CONVERT 3 BINARY-CODED DIGITS
182 0047          ;
183 0047 174D A AEM10:  .WORD   EM10        ; ROUTINE TO MULTIPLY AN INTEGER BY 10
184 0048 1752 A AEM24:  .WORD   EM24        ; ROUTINE TO MULTIPLY AN INTEGER BY 24
185 0049          ;
186 0049 1340 A AERE0:  .WORD   ERE0        ; ERROR RECORDING ROUTINE
187 004A 1349 A AERE1:  .WORD   ERE1        ; ERROR RECORDING ROUTINE
188 004B 1352 A AERE2:  .WORD   ERE2        ; ERROR RECORDING ROUTINE
```

```
189 004C 135B A AERE3:  .WORD  ERE3   ; ERROR RECORDING ROUTINE
190 004D 1364 A AERE4:  .WORD  ERE4   ; ERROR RECORDING ROUTINE
191 004E 136D A AERE5:  .WORD  ERE5   ; ERROR RECORDING ROUTINE
192 004F 1376 A AERE6:  .WORD  ERE6   ; ERROR RECORDING ROUTINE
193 0050 137F A AERE7:  .WORD  ERE7   ; ERROR RECORDING ROUTINE
194 0051 1388 A AERE8:  .WORD  ERE8   ; ERROR RECORDING ROUTINE
195 0052 1391 A AERE9:  .WORD  ERE9   ; ERROR RECORDING ROUTINE
196 0053 139A A AERE10: .WORD  ERE10  ; ERROR RECORDING ROUTINE
197 0054 13A3 A AERE11: .WORD  ERE11  ; ERROR RECORDING ROUTINE
198 0055 13AC A AERE12: .WORD  ERE12  ; ERROR RECORDING ROUTINE
199 0056        ;
200 0056 1F97 A AXLON1: .WORD  XLON1  ; ROUTINE TO TURN STATUS LIGHT ON
201 0057 1F9A A AXLOF1: .WORD  XLOF1  ; ROUTINE TO TURN STATUS LIGHT OFF
202 0058 1F9D A AXLON2: .WORD  XLON2  ; ROUTINE TO TURN STATUS LIGHT ON
203 0059 1FA0 A AXLOF2: .WORD  XLOF2  ; ROUTINE TO TURN STATUS LIGHT OFF
204 005A        ;
205 005A        ;          ... SELF TEST ENTRY POINTS ...
206 005A 1E0C A ASMAD:  .WORD  SMAD   ; RAM ADDRESS TEST
```

```
209 005B                    .LIST   LBP
210 005B 0068 A             .=X'68
211 0068                ;
212 0068                ; DATA INPUT CPC DATA BASE POINTERS
213 0068                ;
214 0068 3100 A ABBUF:  .WORD   BBUF            ; DATA INPUT CPC DATA BASE
215 0069 3100 A ABDIN:  .WORD   BDIN            ; DISCRETE INPUT WORD
216 006A 32BE A ABERR:  .WORD   BERR            ; ERROR BUFFER
217 006B                ;
218 006B                ; DATA OUTPUT CPC DATA BASE POINTERS
219 006B                ;
220 006B 3119 A AXBUF:  .WORD   XBUF            ; DATA OUTPUT CPC DATA BASE
221 006C 32BF A AXERR:  .WORD   XERR            ; ERROR BUFFER
222 006D                ;
223 006D                ; PREEMPT CPC DATA BASE POINTERS
224 006D                ;
225 006D 311E A APBUF:  .WORD   PBUF            ; PREEMPT CPC DATA BASE
226 006E 311E A APIP:   .WORD   PIP                     ; PREEMPT IN PROGRESS FLAG
227 006F 311F A APTEM:  .WORD   PTEM                    ; TEMPORARY STORAGE LOCATION
228 0070 3120 A APTCP:  .WORD   PTCP            ; ADDRESS OF PRIMARY CONTRACTIBLE INT
229 0071 3121 A APTCS:  .WORD   PTCS            ; ADDRESS OF SECONDARY CONTRACTIBLE INT
230 0072 3122 A APTEI:  .WORD   PTEI            ; ADDRESS OF EXTENDIBLE INT
231 0073 3123 A APMIF:  .WORD   PMIF                    ; PREEMPT MONITOR INDEX FLAG
232 0074 3126 A APFREQ: .WORD   PFREQ                   ; PREEMPT REQUEST FLAG
233 0075 3127 A APMDT:  .WORD   PMDT                    ; PREEMPT MINIMUM DURATION TIMER
234 0076 312B A APEMAX: .WORD   PEMAX           ; MAXIMUM EXTENSION FOR PREEMPT
235 0077 32C4 A APERR:  .WORD   PERR            ; PREEMPT ERROR BUFFER
236 0078 3128 A APTAVT: .WORD   PTAVT           ; UNEXPIRED PREEMPT EXTENSION TIME
237 0079 3129 A APTITE: .WORD   PTITE           ; DISP OF INT TO RECEIVE UNEXPIRED
238 007A                                        ;    PREEMPT EXTENSION TIME
239 007A 312A A APTPMT: .WORD   PTPMT           ; BASE ADDRESS OF MASTER TIMING PLAN
240 007B                ;
241 007B 3130 A AVCBT:  .WORD   VCBT            ; TIMER FOR RECENT BUS CLASS.
242 007C 3131 A AVCVC:  .WORD   VCVC            ; VEHICLE COUNTERS (ONE PER DETECTOR)
243 007D 3165 A AVCLOG: .WORD   VCLOG                   ; BEGINNING LOCATION OF VEHICLE LOG
244 007E 312F A AVCDT:  .WORD   VCDT                    ; LOCATION OF DETECTOR INDEX WORD
245 007F 3151 A AVCDB:  .WORD   VCDB            ; TEMPORARY DATA BASE
246 0080 3000 A AVCSDB: .WORD   VCSDB           ; BUS DETECTOR SIGNATURE DATA BASE
247 0081                ;
248 0081                ;TIMING PLAN CPC DATA BASE POINTERS
249 0081                ;
250 0081 3171 A ATBUF:  .WORD   TBUF                    ; BEGINNING LOC OF TIMING PLAN DATA BASE
251 0082 3171 A ATCIWC: .WORD   TCIWC           ; CURRENT COMMAND INDEX WORD
252 0083 3172 A ATCDS:  .WORD   TCDS                    ; CURRENT DIAL SETTING--1, 2, OR 3
253 0084 3173 A ATIN:   .WORD   TIN                     ;CURRENT INTERVAL INDEX
254 0085 3174 A ATINT:  .WORD   TINT                    ; CURRENT INTERVAL TIMER
255 0086 3175 A ATFST:  .WORD   TFST                    ; WAIT FOR SYNC START FLAG
256 0087 3176 A ATEOC:  .WORD   TEOC                    ; END OF CYCLE FLAG
257 0088 3177 A ATEMP:  .WORD   TEMP                    ; TEMPORARY LOCATION
258 0089 3178 A ATDWL:  .WORD   TDWL                    ; MAXIMUM WAIT FOR SYNC START DWELL
259 008A 008A A ATPP:   .WORD   ATPP            ; POINTER TO MASTER TP POINTERS
260 008B                                        ;    (MUST PRECEDE ATPM1,ATPM2,ATPM3)
```

```
008B 317B A ATPM1:  .WORD  TPM1   ; BEGINNING LOC OF MASTER TP, DIAL 1
008C 318F A ATPM2:  .WORD  TPM2   ; BEGINNING LOC OF MASTER TP, DIAL 2
008D 31A3 A ATPM3:  .WORD  TPM3   ; BEGINNING LOC OF MASTER TP, DIAL 3
008E 31B7 A ATNTP:  .WORD  TNTP   ; BEGINNING LOC OF NEXT TP TABLE
008F 31CB A ATCTP:  .WORD  TCTP   ; BEGINNING LOC OF CURRENT TP TABLE
0090 32D9 A ATERR:  .WORD  TERR   ; TIMING PLAN CPC ERROR FLAGS, COUNTERS,
0091                              ; DATA
0091        ;
0091        ; TIMING PLAN TABLES POINTERS
0091        ;
0091 18D3 A ATSCI:  .WORD  TSCI   ; SYNC.CONTRACTIBLE INTERVAL
0092 18D4 A ATHI:   .WORD  THI    ; TRANSFER/HOLD INTERVAL
0093 18D5 A ATAI1:  .WORD  TAI1   ; FIRST ADJ INT INDEX
0094 18D6 A ATAI2:  .WORD  TAI2   ; SECOND ADJ INT INDEX
0095 18D7 A ATIMX:  .WORD  TIMX   ; MAXIMUM VALUE FOR TIN
0096 18D8 A ATECI:  .WORD  TECI   ; BEGINNING LOC OF EXTENDIBLE/
0097                              ; CONTRACTIBLE INTERVAL DATA TABLES
0097 1918 A ATCIW:  .WORD  TCIW   ; BEGINNING LOC OF COMMAND INDEX WORD
0098                              ; TABLE
0098 192C A ATPTAB: .WORD  TPTAB  ; BEGINNING LOC OF TIMING PLAN TABLES
0099 1967 A ATPE:   .WORD  TPE    ; END LOCATION OF TIMING PLAN TABLE ROM
009A        ;
009A        ;EXECUTIVE CPC DATA BASE POINTERS
009A        ;
009A 31DF A AEBUF:  .WORD  EBUF   ; BEGINNING LOC OF EXECUTIVE CPC DATA
009B                              ; BASE
009B 31E1 A AETCY:  .WORD  ETCY   ; TIMING PLAN CYCLE TIMER
009C 31E0 A AETST:  .WORD  ETST   ; WAIT FOR SYNC START DWELL TIMER
009D 31E4 A AEPST:  .WORD  EPST   ; START PULSE INDICATOR
009E 31E5 A AEPDT:  .WORD  EPDT   ; START PULSE DURATION TIMER
009F 32DE A AEERR:  .WORD  EERR   ; ERROR BUFFER
00A0        ;
00A0 31E6 A AECMF:  .WORD  ECMF   ; TEMPORARY FLAG FOR ECOM
00A1        ;
00A1 31E7 A AESYD:  .WORD  ESYD   ; SYNCH DATA BUFFER
00A2        ;
00A2 31F9 A AER123: .WORD  ER123  ; RESET SELECT INTERNAL VALUE
00A3 31FA A AERNO:  .WORD  ERNO   ; RESET SELECT DATA BUFFER
00A4        ;
00A4 31FD A AED23:  .WORD  ED23   ; DIAL SELECT INTERNAL VALUE
00A5 31FE A AEDNO:  .WORD  EDNO   ; DIAL SELECT DATA BUFFER
00A6 3201 A AEDMIT: .WORD  EDMIT  ; DETECTOR MONITORING INTERVAL TIMER
00A7 3202 A AEDETT: .WORD  EDETT  ; TEMP STORAGE FOR DET MONITORING
00A8 333C A AEDERR: .WORD  EDERR  ; DETECTOR ERROR BUFFER
00A9 3C00 A AERAM:  .WORD  X'3C00 ; FIRST LOCATION IN FREE RAM (3C00-3FFF)
```

```
 308 00AA                    .LIST   LBP
 309 00AA          ;
 310 00AA 3203 A AESAV:  .WORD   ESAV          ; GENERAL INTERRUPT SAVE AREA POINTERS
 311 00AB 3205 A AESV2:  .WORD   ESV2
 312 00AC 3207 A AESVF:  .WORD   ESVF
 313 00AD 3218 A AESVS:  .WORD   ESVS
 314 00AE          ;
 315 00AE 3219 A ACSAV:  .WORD   CSAV            ; SNAPSHOT SAVE AREA
 316 00AF 321B A ACSV2:  .WORD   CSV2
 317 00B0 321D A ACSVF:  .WORD   CSVF
 318 00B1 322E A ACSVS:  .WORD   CSVS
 319 00B2          ;
 320 00B2 3231 A AISV2:  .WORD   ISV2          ; STACK FULL SAVE AREA POINTERS
 321 00B3 3233 A AISTK1: .WORD   ISTK1
 322 00B4 3234 A AISTK2: .WORD   ISTK2
 323 00B5 3235 A AISTK3: .WORD   ISTK3
 324 00B6 3236 A AISTK4: .WORD   ISTK4
 325 00B7          ;
 326 00B7 3245 A AEDSV2: .WORD   EDSV2         ; DISCONNECT SAVE AREA POINTERS
 327 00B8 3247 A AEDSVF: .WORD   EDSVF
 328 00B9 3258 A AEDSVS: .WORD   EDSVS
 329 00BA          ;
 330 00BA 3259 A ACTST:  .WORD   CTST            ; OPTIONAL USE SAVE AREA POINTER
```

```
00BB                    .LIST   LBP
00BB          ;
00BB          ;                    ... PROG. PANEL DATA POINTERS ...
00BB 3103 A ABPPD:   .WORD   BPPD          ; PANEL DISCRETE INPUT WORD
00BC 3104 A ABPD1:   .WORD   BPD1          ; TW SWITCH DATA INPUT     DIAL 1
00BD 3108 A ABPD2:   .WORD   BPD2          ; TW SWITCH DATA INPUT     DIAL 2
00BE 3112 A ABPD3:   .WORD   BPD3          ; TW SWITCH DATA INPUT     DIAL 3
00BF          ;
00BF 3261 A AEPD1:   .WORD   EPD1          ; PANEL WORKING BUFFER,    DIAL 1
00C0 3268 A AEPD2:   .WORD   EPD2          ; PANEL WORKING BUFFER,    DIAL 2
00C1 326F A AEPD3:   .WORD   EPD3          ; PANEL WORKING BUFFER,    DIAL 3
00C2          ;
00C2 3276 A AEPCD1:  .WORD   EPCD1         ; PANEL UPDATE BUFFER,     DIAL 1
00C3 3280 A AEPCD2:  .WORD   EPCD2         ; PANEL UPDATE BUFFER,     DIAL 2
00C4 328A A AEPCD3:  .WORD   EPCD3         ; PANEL UPDATE BUFFER,     DIAL 3
00C5          ;
00C5 3294 A AEPFT:   .WORD   EPFT          ; FIXED INTERVAL SUMS
00C6          ;
00C6 329B A AEPUF:   .WORD   EPUF          ; PANEL UPDATE FLAG
00C7 329C A AEPCF:   .WORD   EPCF          ; PROG PANEL MONITOR COMPUTATION FLAG
00C8 329D A AEPIR:   .WORD   EPIR          ; STATE OF PROG PANEL INPUT RED SWITCH
00C9          ;
00C9          ;                    ... DATA CONSTANTS ...
00C9 146A A AEPCY:   .WORD   EPCY          ; MAXIMUM CYCLE LENGTH
00CA 146B A AEPIN1:  .WORD   EPIN1         ; LIMIT ON MAXIMUM EXTENSION
00CB 146D A AE100:   .WORD   E100          ; PERCENTAGE CONVERSION CONSTANT
00CC 146C A AE10:    .WORD   E10           ; CONSTANT FOR DIVISION
00CD          ;
00CD 3297 A AEP1E:   .WORD   EP1E          ; ERROR FLAG FOR PLAN 1
00CE 3298 A AEP2E:   .WORD   EP2E          ; ERROR FLAG FOR PLAN 2
00CF 3299 A AEP3E:   .WORD   EP3E          ; ERROR FLAG FOR PLAN 3
00D0 329A A AEPDVE:  .WORD   EPDVE         ; ERROR FLAG FOR DATA VALIDITY
00D1          ;
00D1 32EB A AEPEF1:  .WORD   EPEF1         ; PROG. PANEL ERROR DATA FOR PLAN 1
00D2 3306 A AEPEF2:  .WORD   EPEF2         ; PROG. PANEL ERROR DATA FOR PLAN 2
00D3 3321 A AEPEF3:  .WORD   EPEF3         ; PROG. PANEL ERROR DATA FOR PLAN 3
```

```
00D4                .LIST   LBP
00D4          ;
00D4          ;                    ... RAM ADDRESS TEST DATA POINTERS ...
00D4 329E A ASSAD:  .WORD   SSAD          ; RAM ADDRESS TEST START ADDRESS
00D5 329F A ASEAD:  .WORD   SEAD          ; RAM ADDRESS TEST END ADDRESS
00D6          ;
00D6          ; END OF DATA BASE POINTER
00D6          ;
00D6 337C A ADEND:  .WORD   DEND          ; END LOCATION OF DATA BASE
```

BASE PAGE - PERIPHERAL DEVICE ADDRESSES

```
00D7                    .LIST   LBP
00D7 00E0 A             .=X'E0
00E0            ;
00E0            ; PERIPHERAL DEVICE ADDRESSES
00E0            ;
00E0 0080 A ALDU:    .WORD   1*128          ; LDU DEVICE ADDRESS
00E1 0100 A ABLSMU:  .WORD   2*128          ; BLSMU DEVICE ADDRESS
00E2 0180 A ACMU:    .WORD   3*128          ; CMU DEVICE ADDRESS
00E3 0200 A ATCP:    .WORD   4*128          ; TCP DEVICE ADDRESS
00E4 0280 A ASDU:    .WORD   5*128          ; SDU DEVICE ADDRESS
00E5 0300 A ADMU:    .WORD   6*128          ; DMU DEVICE ADDRESS
00E6 0380 A ASDB:    .WORD   7*128          ; SDB DEVICE ADDRESS
00E7 0400 A ARTC:    .WORD   8*128          ; RTC DEVICE ADDRESS
00E8            ;
00E8 1F80 A ACCU:    .WORD   63*128         ; CCU DEVICE ADDRESS
00E9            ;
00E9            ;                   ... PERIPHERAL DEVICE ADDRESSES FOR PROG. PANEL ...
00E9 0500 A APPDU:   .WORD   10*128     ; DISCRETES
00EA 0580 A APP1U:   .WORD   11*128     ; PLAN 1 PARAMETERS
00EB 0600 A APP2U:   .WORD   12*128     ; PLAN 2 PARAMETERS
00EC 0680 A APP3U:   .WORD   13*128     ; PLAN 3 PARAMETERS
```

```
00ED                    .LIST   LBP
00FD 00F0 A             .=X'F0
00F0            ;
00F0            ;                       ... COMMON HEXADECIMAL CONSTANTS ...
00F0 7FFF A H7FFF:  .WORD   X'7FFF          ; HEX CONSTANT (BIT 15 RESET)
00F1 3000 A H3000:  .WORD   X'3000          ; DATA BASE START ADDRESS
00F2 3FFF A H3FFF:  .WORD   X'3FFF          ; BITS 15, 14 RESET
00F3                                        ; RAM END ADDRESS
00F3 EC00 A HECC0:  .WORD   X'EC00          ; START ADDRESS OF UNUSED CCU RAM
00F4 ECCF A HECCF:  .WORD   X'ECCF          ; END ADDRESS OF UNUSED CCU RAM
00F5 0200 A H0200:  .WORD   X'0200          ; RTS DO INSTRUCTION
00F6 3BFF A H3BFF:  .WORD   X'3BFF          ; END ADDRESS FOR RTC RAM ADDRESS TEST
```

```
417 00F7                .PAGE
418 00F7                .ASECT
419 00F7 1000 A         .=X'1000
420 1000                ;
```

```
1000                .LIST   LER
1000        ;
1000        ;                   ... E X E C U T I V E   C P C ...
1000        ;
1000        ;                   ... RESTART INITIALIZATION ROUTINE ...
1000        ERSI:
1000 1860 A         BOC     CCSF,ESTK       ; TEST FOR STACKFULL LINE = 1
1001 8CE7 A         LD      AC3,ARTC
1002 4C00 A         LI      AC0,C0
1003 0643 A         ROUT    FC43            ; ASSURE RTC STOPPED
1004 0641 A         ROUT    FC41            ; ASSURE RTC INTERRUPT PULSE RESET
1005        ;
1005 0980 A         PFLG    INEN            ; DISABLE INTERRUPTS
1006        ;
1006        ;                   ... INITIALIZE CCU SERVICING ROUTINE ...
1006 2C22 A         JSR     @AECCU
1007        ;
1007        ;                   ... INITIAL RAM ADDRESS TEST ...
1007 0714 A         CLRST   D4              ; CLEAR MEMORY ADDRESS TEST STATUS FLAG
1008 2C23 A         JSR     @AEMAD
1009        ;
1009 0715 A         CLRST   D5
100A 0716 A         CLRST   D6
100B 0717 A         CLRST   D7
100C        ;
100C 80F5 A         LD      AC0,H0200       ; RTS DO INSTRUCTION
100D B0A9 A         ST      AC0,@AERAM      ; STORE IN FIRST LOCATION OF FREE RAM
100E                                        ;   (ENTRY POINT VIA COTF)
100E        ;
100E        ;                   ... RESET CONFLICT MONITOR ...
100E 8CE2 A         LD      AC3,ACMU
100F 4C00 A         LI      AC0,C0
1010 0641 A         ROUT    FC41
1011        ;
1011        ;                   ... INITIALIZE CPU STACK ...
1011 4EF0 A         LI      AC2,CM16
1012 4700 A         PULL    AC3             ; PULL 16 WORDS FROM STACK
1013 4A01 A         AISZ    AC2,C1
1014 21FD A         JMP     .-2
1015        ;
1015        ;                   ... INITIALIZE STACK FOR BACKGROUND ROUTINES ...
1015 4FFF A         LI      AC3,CM1
1016 4300 A         PUSH    AC3
1017 4300 A         PUSH    AC3             ; PUSH TWO PROTECTION WORDS OF ALL ONES
1018        ;
1018 889A A         LD      AC2,AEBUF
1019 A200 A         ST      AC0,D0(AC2)     ; ZERO RTC COMPUTE FLAG
101A        ;
101A        ;                               ZERO DATA BASE FROM VCSDB TO DEND
101A 4C00 A         LI      AC0,C0
101B 8880 A         LD      AC2,AVCSDB
101C A200 A EZDB:   ST      AC0,(AC2)
```

```
101D 4A01 A        AISZ   AC2,C1
101E 2100 A        JMP    .+1
101F E8D6 A        SKG    AC2,ADEND
1020 21FB A        JMP    EZDB
1021          ;
1021          ;                                  COPY TIMING PLAN TABLES FROM ROM TO READ/
1021          ;                                  WRITE DATA BASE
1021 8898 A        LD     AC2,ATPTAB
1022 8C8B A        LD     AC3,ATPM1
1023 8200 A ETCPY: LD     AC0,(AC2)
1024 A300 A        ST     AC0,(AC3)
1025 4A01 A        AISZ   AC2,C1
1026 4B01 A        AISZ   AC3,C1
1027 E899 A        SKG    AC2,ATPE
1028 21FA A        JMP    ETCPY
1029          ;
1029 889B A        LD     AC2,AETCY          ; INITIALIZE TIMING PLAN CYCLE TIMER
102A 8115 A        LD     AC0,H0FFF          ;    MINIMUM TO A LARGE VALUE
102B A202 A        ST     AC0,D2(AC2)        ;
102C          ;
102C          ;                                  ... PROGRAMMABLE PANEL ROUTINE ...
102C 2C3F A        JSR    @AEPAN
102D          ;
102D          ;                                  ... READ DISCRETE INPUT WORDS, ETC. ...
102D 2C3C A        JSR    @ABRTC
102E          ;
102E          ;                                  DETERMINE INITIAL DIAL SETTING AND COPY
102E          ;                                  APPROPRIATE MASTER TIMING PLAN INTO CURRENT
102E          ;                                  AND NEXT TIMING PLANS
102E 2C38 A        JSR    @ATDIAL
102F          ;
102F          ;                                  SET INTERVAL COUNTER TO TH INTERVAL
102F 9892 A        LD     AC2,@ATHI
1030 B884 A        ST     AC2,@ATIN
1031          ;
1031          ;                                  SET CURRENT INTERVAL TIMER TO LENGTH OF
1031          ;                                  CURRENT TH INTERVAL
1031 C88F A        ADD    AC2,ATCTP
1032 8600 A        LD     AC1,(AC2)
1033 B485 A        ST     AC1,@ATINT
1034          ;
1034          ;                                  COPY APPROPRIATE COMMAND INDEX WORD INTO
1034          ;                                  BUFFER(TCIWC)
1034 9892 A        LD     AC2,@ATHI
1035 C897 A        ADD    AC2,ATCIW
1036 8600 A        LD     AC1,(AC2)
1037 B482 A        ST     AC1,@ATCIWC
1038          ;
1038          ;                                  FORMAT OUTPUT WORD
1038 2C3B A        JSR    @AXSDC
1039          ;
1039          ;
```

```
1039          ;                    ... INITIALIZATION PSEUDO HALT ...
1039 1DFF A EHLT:   BOC     CCPH,EHLT        ; PSEUDO HALT LOOP
103A          ;
103A          ;
103A 4C01 A         LI      AC0,C1
103B 8CE7 A         LD      AC3,ARTC
103C 0642 A         ROUT    FC42             ; START RTC
103D          ;
103D          ;                    ... INITIATE BACKGROUND PROGRAMS ...
103D 8C24 A         LD      AC3,AEBKD
103E 4300 A         PUSH    AC3
103F          ;
103F 0100 A         RTI     D0               ; RETURN AND ENABLE INTERRUPTS ...
1040          ;
1040          ;                    ... LOCAL DATA ...
1040          ;
1040 0FFF A HOFFF:  .WORD   X'0FFF
1041          ;
```

```
1041                .LIST   LER
1041          ;
1041          ;                    ... GENERAL INTERRUPT SERVICING ROUTINE ...
1041          EGIS:
1041 181F A           BOC     CCSF,ESTK         ; TEST FOR STACK FULL LINE = 1
1042          ;
1042 B8AB A           ST      AC2,@AESV2        ; SAVE ACCUMULATORS
1043 B8AB A           LD      AC2,AESV2
1044 A2FE A           ST      AC0,DM2(AC2)
1045 A6FF A           ST      AC1,DM1(AC2)
1046 AE01 A           ST      AC3,D1(AC2)
1047          ;
1047 4400 A           PULL    AC0               ; SAVE RALU FLAGS
1048 0080 A           PUSHF
1049 5400 A           XCHRS   AC0
104A B0AC A           ST      AC0,@AESVF
104B          ;
104B 4D04 A           LI      AC1,C4            ; SAVE CPU STACK
104C B8AC A           LD      AC2,AESVF         ; STACK FOLLOWS RALU FLAG WORD
104D 4700 A EGIS1:    PULL    AC3               ; PULL STACK FROM TOP TO BOTTOM
104E AE01 A           ST      AC3,D1(AC2)
104F 4700 A           PULL    AC3
1050 AE02 A           ST      AC3,D2(AC2)
1051 4700 A           PULL    AC3
1052 AE03 A           ST      AC3,D3(AC2)
1053 4700 A           PULL    AC3
1054 AE04 A           ST      AC3,D4(AC2)
1055 4A04 A           AISZ    AC2,D4            ; INCREMENT SAVE AREA ADDRESS BY 4
1056 49FF A           AISZ    AC1,CM1           ; DECREMENT PASS COUNTER, SKIP IF DONE
1057 21F5 A           JMP     EGIS1
1058          ;
1058          ;
1058 4C01 A           LI      AC0,C1            ; SAVE SELECT FLAG STATE
1059 58EF A           ROR     AC0,D17
105A 1303 A           BOC     CCB0,EGIS2        ; BRANCH IF SELECT FLAG SET
105B 4C00 A           LI      AC0,C0
105C B0AD A           ST      AC0,@AESVS        ; SELECT FLAG = 0
105D 212D A           JMP     EGIS3
105E 4C01 A EGIS2:    LI      AC0,C1
105F B0AD A           ST      AC0,@AESVS        ; SELECT FLAG = 1
1060 212A A           JMP     EGIS3
1061          ;
1061          ;
1061          ;
1061          ;                    ... CLEAR STACK FULL CONDITION ...
1061          ESTK:
1061 5400 A           XCHRS   AC0
1062 B0B3 A           ST      AC0,@AISTK1       ; SAVE STACK 1
1063 4400 A           PULL    AC0               ; RESTORE AC0
1064 5400 A           XCHRS   AC0
1065 B0B4 A           ST      AC0,@AISTK2       ; SAVE STACK 2
1066 4400 A           PULL    AC0               ; RESTORE AC0
```

```
1067 5400 A          XCHRS   AC0
1068 B0B5 A          ST      AC0,@AISTK3      ; SAVE STACK 3
1069 4400 A          PULL    AC0              ; RESTORE AC0
106A 5400 A          XCHRS   AC0
106B B0B6 A          ST      AC0,@AISTK4      ; SAVE STACK 4
106C 4400 A          PULL    AC0              ; RESTORE AC0
106D            ;
106D B8B2 A          ST      AC2,@AISV2       ; SAVE ACCUMULATORS
106E 89B2 A          LD      AC2,AISV2
106F A2FE A          ST      AC0,DM2(AC2)
1070 A6FF A          ST      AC1,DM1(AC2)
1071 AE01 A          ST      AC3,D1(AC2)
1072            ;
1072 8CE7 A          LD      AC3,ARTC
1073 4C00 A          LI      AC0,C0
1074 0643 A          ROUT    FC43             ; STOP RTC
1075 0980 A          PFLG    INEN             ; DISABLE INTERRUPTS
1076            ;
1076 88B6 A          LD      AC2,AISTK4       ; SAVE STACK 5 - 16
1077 4D03 A          LI      AC1,C3
1078 4700 A ESTK1:   PULL    AC3
1079 AE01 A          ST      AC3,D1(AC2)
107A 4700 A          PULL    AC3
107B AE02 A          ST      AC3,D2(AC2)
107C 4700 A          PULL    AC3
107D AE03 A          ST      AC3,D3(AC2)
107E 4700 A          PULL    AC3
107F AE04 A          ST      AC3,D4(AC2)
1080 4A04 A          AISZ    AC2,C4
1081 49FF A          AISZ    AC1,CM1
1082 21F5 A          JMP     ESTK1
1083            ;
1083 89B2 A          LD      AC2,AISV2        ; RESTORE ACCUMULATORS
1084 82FE A          LD      AC0,DM2(AC2)
1085 86FF A          LD      AC1,DM1(AC2)
1086 8E01 A          LD      AC3,D1(AC2)
1087 98B2 A          LD      AC2,@AISV2
1088            ;
1088 2C2F A          JSR     @AEADC           ; AUTODISCONNECT ROUTINE, NO RETURN
1089            ;
1089 2100 A          JMP     .+1              ; THIS INST SHOULD NEVER BE REACHED
108A 21FE A          JMP     .-1
108B            ;
108B            ;                ... DETERMINE SOURCE OF INTERRUPT ...
108B            EGIS3:
108B            ;
108B            ;                ... RTC IS THE ONLY INTERRUPT PRESENTLY ASSIGNED ...
108B            ;
108B            ;                ... REAL TIME CLOCK INTERRUPT SERVICING ROUTINE ...
108B 889A A ERTC:    LD      AC2,AEBUF        ; CHECK FOR COMPLETION OF PREVIOUS
108C 8200 A          LD      AC0,D0(AC2)      ;    RTC COMPUTE SEQUENCE
108D 110A A          BOC     CCEZ,ERTC1       ; BRANCH IF NO ERROR
```

```
651 108E          ;
652 108E 889F A          LD      AC2,AEERR
653 108F A202 A          ST      AC0,D2(AC2)      ; STORE RTC COMPUTE FLAG, EERRO
654 1090 8201 A          LD      AC0,D1(AC2)
655 1091 4801 A          AISZ    AC0,C1           ; INCREMENT ERROR COUNTER
656 1092 2100 A          JMP     .+1
657 1093 A201 A          ST      AC0,D1(AC2)
658 1094 8200 A          LD      AC0,D0(AC2)
659 1095 0720 A          SETBIT  D0               ; SET ERROR FLAG BIT 0
660 1096 A200 A          ST      AC0,D0(AC2)
661 1097 2425 A          JMP     @AECER           ; RTC ERROR RESPONSE ROUTINE
662 1098          ;
663 1098 4C01 A ERTC1:   LI      AC0,C1           ; SET RTC COMPUTE FLAG
664 1099 A200 A          ST      AC0,D0(AC2)
665 109A          ;                   ... INITIALIZE STACK FOR RTC SEQUENCE ...
666 109A          ;
667 109A 4FFF A          LI      AC3,CM1
668 109B 4300 A          PUSH    AC3
669 109C 4300 A          PUSH    AC3              ; PUSH TWO PROTECTION WORDS OF ALL ONES
670 109D          ;
671 109D 2C26 A          JSR     @AESCH           ; CALL SCHEDULING ROUTINE
672 109E          ;
673 109E 4700 A          PULL    AC3              ; PULL TWO PROTECTION WORDS FROM STACK
674 109F 4700 A          PULL    AC3
675 10A0          ;
676 10A0          ;                   ... ENABLE OTHER INTERRUPTS ...
677 10A0          ;
678 10A0          ;                   ... NO OTHER INTERRUPTS PRESENTLY ASSIGNED ...
679 10A0 889A A          LD      AC2,AEBUF        ; RESET RTC COMPUTE FLAG
680 10A1 4C00 A          LI      AC0,C0
681 10A2 A200 A          ST      AC0,D0(AC2)
682 10A3          ;
683 10A3          ;                   ... RESTORE COMPUTATIONAL ENVIRONMENT ...
684 10A3 90AD A          LD      AC0,@AESVS       ; RESTORE SELECT FLAG
685 10A4 1102 A          BOC     CCEZ,EGIS4
686 10A5 0A00 A          SFLG    SELF
687 10A6 2101 A          JMP     .+2
688 10A7 0A80 A EGIS4:   PFLG    SELF
689 10A8          ;
690 10A8          ;
691 10A8 4D04 A          LI      AC1,C4           ; RESTORE CPU STACK
692 10A9 88AD A          LD      AC2,AESVS        ; STACK PRECEDES SELECT FLAG WORD
693 10AA 4AFC A EGIS5:   AISZ    AC2,DM4          ; DECREMENT SAVE AREA ADDRESS BY 4
694 10AB 8E03 A          LD      AC3,D3(AC2)      ; PUSH STACK FROM BOTTOM TO TOP
695 10AC 4300 A          PUSH    AC3
696 10AD 8E02 A          LD      AC3,D2(AC2)
697 10AE 4300 A          PUSH    AC3
698 10AF 8E01 A          LD      AC3,D1(AC2)
699 10B0 4300 A          PUSH    AC3
700 10B1 8E00 A          LD      AC3,D0(AC2)
701 10B2 4300 A          PUSH    AC3
702 10B3 49FF A          AISZ    AC1,CM1          ; DECREMENT PASS COUNTER, SKIP IF DONE
```

```
703 10B4 21F5 A        JMP     EGIS5
704 10B5          ;
705 10B5 90AC A        LD      AC0,@AESVF      ; RESTORE RALU FLAGS
706 10B6 5400 A        XCHRS   AC0
707 10B7 0280 A        PULLF
708 10B8 4000 A        PUSH    AC0
709 10B9          ;
710 10B9          ;                              RESTORE ACCUMULATORS
711 10B9 88AB A        LD      AC2,AESV2
712 10BA 82FE A        LD      AC0,DM2(AC2)
713 10BB 86FF A        LD      AC1,DM1(AC2)
714 10BC 8E01 A        LD      AC3,D1(AC2)
715 10BD 98AB A        LD      AC2,@AESV2
716 10BE          ;
717 10BE 0100 A        RTI                     ; RETURN AND ENABLE INTERRUPTS
718 10BF          ;
719 10BF          ;                 ... RTC ERROR, PREVIOUS SEQUENCE INCOMPLETE ...
720 10BF          ECER:
721 10BF 2C2F A        JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
722 10C0          ;
723 10C0 2100 A        JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
724 10C1 21FE A        JMP     .-1
```

```
10C2          .LIST   LER
10C2          ;
10C2          ;                 ... COMPUTER CONTROL UNIT INITIALIZATION ...
10C2          ECCU:
10C2          ;
10C2          ;                 ... TEST FOR CCU IN SYSTEM ...
10C2 1C01 A           BOC     CCCU,.+2        ; BRANCH IF CCU IS IN SYSTEM
10C3 2115 A           JMP     ECCUR           ; JUMP IF CCU NOT IN SYSTEM
10C4          ;
10C4 9115 A           LD      AC0,@ACCUI      ; CHECK FIRST INST IN CCU ROUTINE
10C5 F115 A           SKNE    AC0,ECCUI       ; SKIP IF CCU NOT IN SYSTEM
10C6 2101 A           JMP     .+2
10C7 2111 A           JMP     ECCUR           ; JUMP TO RETURN
10C8          ;
10C8          ;                 ... INITIALIZE CCU RAM ...
10C8 4C00 A           LI      AC0,C0
10C9 4D30 A           LI      AC1,C48         ; LENGTH OF CCU RAM = 48
10CA 8913 A           LD      AC2,ACWAP       ; INITIALIZE ADDRESS POINTER
10CB          ;
10CB A2F4 A ECCU1:    ST      AC0,DM12(AC2)   ; CLEAR RAM
10CC 4A01 A           AISZ    AC2,C1          ; INCREMENT ADDRESS POINTER
10CD 49FF A           AISZ    AC1,CM1         ; DECREMENT COUNTER, SKIP IF DONE
10CE 21FC A           JMP     ECCU1           ; CONTINUE LOOP
10CF          ;
10CF 890E A           LD      AC2,ACWAP       ; INITIALIZE ADDRESS POINTER
10D0 7AF9 A           ISZ     DM7(AC2)        ; SET STATE WORD = +1
10D1 7E0C A           DSZ     D12(AC2)        ; SET BREAKPOINT ADDRESS 0 = -1
10D2 7E0E A           DSZ     D14(AC2)        ; SET BREAKPOINT ADDRESS 1 = -1
10D3          ;
10D3 810B A           LD      AC0,AECCUR      ; RETURN ADDRESS FROM CCU ROUTINE
10D4 A203 A           ST      AC0,D3(AC2)     ; STORE IN TEMPORARY PC LOC IN CCU RAM
10D5          ;
10D5 8107 A           LD      AC0,ACRAM       ; BASE ADDRESS OF CCU RAM WORK AREA
10D6 A2FA A           ST      AC0,CM6(AC2)    ; INITIALIZE PUSH/PULL POINTER
10D7          ;
10D7 4C00 A           LI      AC0,C0          ; AC0 = 0, INITIAL DISPLAY
10D8 2503 A           JMP     @ACDISP         ; TO CCU ROUTINE, RETURN IS TO ECCUR
10D9          ;
10D9 0200 A ECCUR:    RTS     D0              ; RETURN FROM ECCU
10DA          ;
10DA          ;                 ... LOCAL DATA ...
10DA EF00 A ACCUI:    .WORD   X'EF00          ; LOC OF FIRST INST IN CCU ROUTINE
10DB 86F7 A ECCUI:    .WORD   X'86F7          ; FIRST INST IN CCU ROUTINE
10DC EF84 A ACDISP:   .WORD   X'EF84          ; CCU ROM + X'84
10DD ECD0 A ACRAM:    .WORD   X'ECD0          ; FIRST LOC IN CCU RAM WORK AREA
10DE ECDC A ACWAP:    .WORD   X'ECDC          ; CCU RAM WORK AREA + X'C
10DF 10D9 A AECCUR:   .WORD   ECCUR           ; RETURN ADDRESS FROM CCU INITIALIZATION
```

```
10E0                    .LIST   LER
10E0            ;
10E0            ;                   ... INITIALIZATION RAM ADDRESS TEST ...
10E0            EMAD:
10E0            ;
10E0            ;                   ... TEST 4K STATIC RAM ...
10E0            ;                       (CAN ONLY TEST ADDRESS -NOT- USED BY DATA BASE)
10E0 80D6 A             LD      AC0,ADEND       ; START ADDRESS (PAST DATA BASE)
10E1 B0D4 A             ST      AC0,@ASSAD
10E2 80F2 A             LD      AC0,H3FFF       ; END ADDRESS
10E3 B0D5 A             ST      AC0,@ASEAD
10E4            ;
10E4 2C5A A             JSR     @ASMAD
10E5 2109 A             JMP     EMADE           ; ERROR RETURN, (AC2) = CONTENTS
10E6                                            ;               (AC3) = ADDRESS
10E6 2100 A             JMP     .+1             ; NO ERROR RETURN
10E7            ;
10E7            ;                   ... TEST CCU RAM ON IMP/16C CARD ...
10E7            ;                       ( CAN ONLY TEST ADDRESSES -NOT- USED BY CCU)
10E7 80F3 A             LD      AC0,HEC00       ; START ADDRESS
10E8 B0D4 A             ST      AC0,@ASSAD
10E9 80F4 A             LD      AC0,HECCF       ; END ADDRESS
10EA B0D5 A             ST      AC0,@ASEAD
10EB            ;
10EB 2C5A A             JSR     @ASMAD
10EC 2102 A             JMP     EMADE           ; ERROR RETURN, (AC2) = CONTENTS
10ED                                            ;               (AC3) = ADDRESS
10ED 2100 A             JMP     .+1             ; NO ERROR RETURN
10EE            ;
10EE 2103 A             JMP     EMADR
10EF            ;
10EF 2C2F A     EMADE:  JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
10F0            ;
10F0 2100 A             JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
10F1 21FE A             JMP     .-1
10F2            ;
10F2 0200 A     EMADR:  RTS     DO              ; RETURN FROM EMAD
10F3            ;
```

```
10F3          .LIST   LER
10F3      ;
10F3      ;                 ... RTC SCHEDULING ROUTINE ...
10F3      ESCH:
10F3 8CE7 A         LD      AC3,ARTC
10F4 4C00 A         LI      AC0,C0          ; CLEAR AC0
10F5 0641 A         ROUT    FC41            ; RESET RTC INTERRUPT PULSE
10F6      ;
10F6 0900 A         SFLG    INEN            ; ENABLE INTERRUPTS
10F7      ;
10F7 1E01 A         BOC     CCCF,.+2        ; BRANCH IF COTF JUMP CONDITION SET
10F8 2101 A         JMP     .+2
10F9 2C3D A         JSR     @ACOTF          ; CALL CATCH ON THE FLY ROUTINE
10FA      ;
10FA 2C39 A         JSR     @AXRTC
10FB 2C3C A         JSR     @ABRTC
10FC 2C27 A         JSR     @AEDIS
10FD 2C28 A         JSR     @AECOM
10FE 2C29 A         JSR     @AESYN
10FF 2C2A A         JSR     @AEMIC
1100 2C2B A         JSR     @AETIM
1101 2C2E A         JSR     @AEDET
1102 2C30 A         JSR     @APMON
1103 2C35 A         JSR     @ATPEX
1104 2C3B A         JSR     @AXSDC
1105 2C2D A         JSR     @AESDW
1106 2C40 A         JSR     @AEPPM
1107      ;
1107 2105 A         JMP     ERTS
1108      ;
1108 8905 A         LD      AC2,EDLAY
1109 5C76 A         SHL     AC0,D118
110A 5C74 A         SHL     AC0,D116
110B 4A01 A         AISZ    AC2,C1
110C 21FC A         JMP     .-3
110D      ;
110D 0200 A ERTS:   RTS     DO
110E      ;                           VARIABLE DELAY
110E FFCE A EDLAY:  .WORD   -50
110F      ;
110F      ;
```

```
110F                .LIST   LER
110F        ;
110F        ;                   ... AUTODISCONNECT ROUTINE ...
110F        EADC:
110F B8B7 A         ST      AC2,@AEDSV2     ; SAVE ACCUMULATORS
1110 B8B7 A         LD      AC2,AEDSV2
1111 A2FE A         ST      AC0,DM2(AC2)
1112 A6FF A         ST      AC1,DM1(AC2)
1113 AE01 A         ST      AC3,D1(AC2)
1114        ;
1114 8CE7 A         LD      AC3,ARTC
1115 4C00 A         LI      AC0,C0
1116 0643 A         ROUT    FC43            ; STOP RTC
1117 0641 A         ROUT    FC41            ; ASSURE RTC INTERRUPT PULSE RESET
1118 0980 A         PFLG    INEN            ; DISABLE INTERRUPTS
1119        ;
1119 4400 A         PULL    AC0             ; SAVE RALU FLAGS
111A 0080 A         PUSHF
111B 5400 A         XCHRS   AC0
111C B0B8 A         ST      AC0,@AEDSVF
111D        ;
111D 4D04 A         LI      AC1,C4          ; SAVE CPU STACK
111E B8B8 A         LD      AC2,AEDSVF      ; STACK FOLLOWS RALU FLAG WORD
111F 4700 A EADC1:  PULL    AC3             ; PULL STACK FROM TOP TO BOTTOM
1120 AE01 A         ST      AC3,D1(AC2)
1121 4700 A         PULL    AC3
1122 AF02 A         ST      AC3,D2(AC2)
1123 4700 A         PULL    AC3
1124 AE03 A         ST      AC3,D3(AC2)
1125 4700 A         PULL    AC3
1126 AE04 A         ST      AC3,D4(AC2)
1127 4A04 A         AISZ    AC2,D4          ; INCREMENT SAVE AREA ADDRESS BY 4
1128 49FF A         AISZ    AC1,CM1         ; DECREMENT PASS COUNTER, SKIP IF DONE
1129 21F5 A         JMP     EADC1
112A        ;
112A        ;
112A 4C01 A         LI      AC0,C1          ; SAVE SELECT FLAG STATE
112B 58EF A         ROR     AC0,D17
112C 1303 A         BOC     CCB0,EADC2      ; BRANCH IF SELECT FLAG SET
112D 4C00 A         LI      AC0,C0
112E B0B9 A         ST      AC0,@AEDSVS     ; SELECT FLAG = 0
112F 2102 A         JMP     EADC3
1130 4C01 A EADC2:  LI      AC0,C1
1131 B0B9 A         ST      AC0,@AEDSVS     ; SELECT FLAG = 1
1132        ;
1132        EADC3:
1132        ;
1132 2C2D A         JSR     @AESDW          ; CONSTRUCT STATUS DISPLAY WORD
1133        ;
1133 906B A         LD      AC0,@AXBUF      ; FETCH SDW
1134 0751 A         SKBIT   D1              ; CHECK EXTERNAL DISCONNECT BIT
1135 2101 A         JMP     .+2             ; JUMP IF NOT EXTERNAL
```

```
911 1136 2104 A          JMP     EADC4          ; JUMP IF EXTERNAL
912 1137 0720 A          SETBIT  D0             ; SET DISCONNECT BIT
913 1138 B06B A          ST      AC0,@AXBUF
914 1139 8CE3 A          LD      AC3,ATCP       ; OUTPUT CONTROL PULSE FOR
915 113A 0643 A          ROUT    FC43           ;    INTERNAL DISCONNECT TO TCP
916 113B        ;
917 113B 8CE4 A EADC4:   LD      AC3,ASDU       ; OUTPUT SDW TO SDU
918 113C 906B A          LD      AC0,@AXBUF     ; FETCH SDW
919 113D 0601 A          ROUT    FC1
920 113E        ;
921 113E        ;
922 113E 3081 A          NOP                    ; TERMINAL WAIT LOOP, NO RETURN
923 113F 3081 A          NOP
924 1140 21FD A          JMP     .-2
925 1141 0000 A          HALT                   ; THIS HALT SHOULD NEVER BE REACHED
926 1142 0000 A          HALT
```

```
1143                    .LIST   LER
1143          ;
1143          ;                 ... TIME OUT FOR DEBUG ROUTINE ...
1143          ;*****************************************************************
1143          ;
1143          COTF:
1143 B8AF A           ST      AC2,@ACSV2          ; SAVE ACCUMULATORS
1144 88AF A           LD      AC2,ACSV2
1145 A2FE A           ST      AC0,DM2(AC2)
1146 A6FF A           ST      AC1,DM1(AC2)
1147 AE01 A           ST      AC3,D1(AC2)
1148          ;
1148 4400 A           PULL    AC0                 ; SAVE RALU FLAGS
1149 0080 A           PUSHF
114A 5400 A           XCHRS   AC0
114B B0B0 A           ST      AC0,@ACSVF
114C          ;
114C 4D04 A           LI      AC1,C4              ; SAVE CPU STACK
114D 88B0 A           LD      AC2,ACSVF           ; STACK FOLLOWS RALU FLAG WORD
114E 4700 A COTF1:    PULL    AC3                 ; PULL STACK FROM TOP TO BOTTOM
114F AE01 A           ST      AC3,D1(AC2)
1150 4700 A           PULL    AC3
1151 AE02 A           ST      AC3,D2(AC2)
1152 4700 A           PULL    AC3
1153 AE03 A           ST      AC3,D3(AC2)
1154 4700 A           PULL    AC3
1155 AE04 A           ST      AC3,D4(AC2)
1156 4A04 A           AISZ    AC2,D4              ; INCREMENT SAVE AREA ADDRESS BY 4
1157 49FF A           AISZ    AC1,CM1             ; DECREMENT PASS COUNTER, SKIP IF DONE
1158 21F5 A           JMP     COTF1
1159          ;-----------------------------------------------------------------
1159          ;
1159 4C01 A           LI      AC0,C1              ; SAVE SELECT FLAG STATE
115A 58EF A           ROR     AC0,D17
115B 1303 A           BOC     CCB0,COTF2          ; BRANCH IF SELECT FLAG SET
115C 4C00 A           LI      AC0,C0
115D B0B1 A           ST      AC0,@ACSVS          ; SELECT FLAG = 0
115E 2102 A           JMP     COTF3
115F 4C01 A COTF2:    LI      AC0,C1
1160 B0B1 A           ST      AC0,@ACSVS          ; SELECT FLAG = 1
1161          ;
1161          ;                                   STOP THE CLOCK
1161          COTF3:
1161 8CE7 A           LD      AC3,ARTC
1162 4C00 A           LI      AC0,C0
1163 0643 A           ROUT    FC43                ; STOP RTC
1164 0641 A           ROUT    FC41                ; RESET RTC INTERRUPT PULSE
1165          ;
1165          ;-----------------------------------------------------------------
1165 3081 A           NOP                         ; INSERT DEBUG HALT OR SNAP HERE
1166 3081 A           NOP
1167 2CA9 A           JSR     @AERAM              ; FIRST LOCATION IN FREE RAM
```

```
1168 3081 A          NOP
1169 3081 A          NOP
116A 3081 A          NOP
116B          ;------------------------------------------------------------
116B          ;
116B          ;                                   START THE CLOCK
116B 8CE7 A          LD      AC3,ARTC
116C 4C00 A          LI      AC0,C0
116D 0642 A          ROUT    FC42                    ; START RTC
116E          ;------------------------------------------------------------
116E          ;                 ... RESTORE COMPUTATIONAL ENVIRONMENT ...
116E          ;
116E 90B1 A          LD      AC0,@ACSVS      ; RESTORE SELECT FLAG
116F 1102 A          BOC     CCEZ,COTF4
1170 0A00 A          SFLG    SELF
1171 2101 A          JMP     .+2
1172 0A80 A COTF4:   PFLG    SELF
1173          ;
1173          ;------------------------------------------------------------
1173 4D04 A          LI      AC1,C4          ; RESTORE CPU STACK
1174 88B1 A          LD      AC2,ACSVS       ; STACK PRECEDES SELECT FLAG WORD
1175 4AFC A COTF5:   AISZ    AC2,DM4         ; DECREMENT SAVE AREA ADDRESS BY 4
1176 8E03 A          LD      AC3,D3(AC2)     ; PUSH STACK FROM BOTTOM TO TOP
1177 4300 A          PUSH    AC3
1178 8E02 A          LD      AC3,D2(AC2)
1179 4300 A          PUSH    AC3
117A 8E01 A          LD      AC3,D1(AC2)
117B 4300 A          PUSH    AC3
117C 8E00 A          LD      AC3,D0(AC2)
117D 4300 A          PUSH    AC3
117E 49FF A          AISZ    AC1,CM1         ; DECREMENT PASS COUNTER, SKIP IF DONE
117F 21F5 A          JMP     COTF5
1180          ;
1180 90B0 A          LD      AC0,@ACSVF      ; RESTORE RALU FLAGS
1181 5400 A          XCHRS   AC0
1182 0280 A          PULLF
1183 4000 A          PUSH    AC0
1184          ;
1184          ;                                   RESTORE ACCUMULATORS
1184 88AF A          LD      AC2,ACSV2
1185 82FE A          LD      AC0,DM2(AC2)
1186 86FF A          LD      AC1,DM1(AC2)
1187 8E01 A          LD      AC3,D1(AC2)
1188 98AF A          LD      AC2,@ACSV2
1189          ;
1189 0200 A          RTS     DO              ; RETURN FROM COTF
```

```
1027 118A                    .PAGE
1028 118A                    .ASECT
1029 118A 1200 A             .=X'1200
1030 1200                    ;
```

```
1200                    .LIST   LES
1200          ;
1200          ;                 ... MONITOR FOR DISCONNECT ...
1200          EDIS:
1200          ;
1200          ;                 ... TEST PROCESSOR STATUS FLAGS ...
1200 0744 A             SKSTF   D4              ; SKIP IF RAM ADDRESS TEST ERROR
1201 2101 A             JMP     .+2
1202 210C A             JMP     EDISH
1203          ;
1203          ;                 ... RESERVE TESTS FOR ADDITIONAL STATUS FLAGS ...
1203 0745 A             SKSTF   D5
1204 2101 A             JMP     .+2
1205 2109 A             JMP     EDISH
1206          ;
1206 0746 A             SKSTF   D6
1207 2101 A             JMP     .+2
1208 2106 A             JMP     EDISH
1209          ;
1209 0747 A             SKSTF   D7
120A 2101 A             JMP     .+2
120B 2103 A             JMP     EDISH
120C          ;
120C 9069 A             LD      AC0,@ABDIN      ; DISCRETE INPUT WORD
120D 075F A             SKBIT   D15             ; SKIP IF DISCONNECT BIT IS SET
120E 0200 A             RTS     D0
120F          ;
120F 2C2F A EDISH:      JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
1210          ;
1210 2100 A             JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
1211 21FE A             JMP     .-1
1212          ;
```

```
1212                .LIST LES
1212          ;
1212          ;                 ... MONITOR COMMAND FEEDBACK DATA FROM LDU ...
1212          ECOM:
1212 4C00 A          LI      AC0,C0
1213 B0A0 A          ST      AC0,@AECMF        ; CLEAR CURRENT RTC INTERVAL ERROR FLAG
1214          ;
1214          ;                 ... COMMAND FEEDBACK TEST FOR MAIN STREET ...
1214 886B A          LD      AC2,AXBUF
1215 8201 A          LD      AC0,D1(AC2)       ; XOCW1 OUTPUT COMMAND
1216 60F2 A          AND     AC0,H3FFF         ; MASK OFF DON'T CARE BITS 15,14
1217 8868 A          LD      AC2,ABBUF
1218 8601 A          LD      AC1,D1(AC2)       ; BCR1 FEEDBACK COMMAND
1219 60F2 A          AND     AC0,H3FFF         ; MASK OFF DON'T CARE BITS 15,14
121A 3482 A          RXOR    AC1,AC0           ; COMPARE, AC1 UNALTERED
121B 110B A          BOC     CCEZ,ECOM1        ; BRANCH IF IDENTICAL (AC0=0)
121C          ;
121C          ;                 ... MAIN STREET COMMAND FEEDBACK ERROR ...
121C 889F A          LD      AC2,AEERR
121D A604 A          ST      AC1,D4(AC2)       ; STORE BCR1 IN ERROR BUFFER, EERR1
121E 8203 A          LD      AC0,D3(AC2)       ; FETCH ERROR COUNTER AND INCREMENT
121F 4801 A          AISZ    AC0,C1
1220 2100 A          JMP     .+1
1221 A203 A          ST      AC0,D3(AC2)
1222 8200 A          LD      AC0,D0(AC2)
1223 0721 A          SETBIT  D1                ; SET ERROR FLAG BIT 1
1224 A200 A          ST      AC0,D0(AC2)
1225 4C01 A          LI      AC0,C1
1226 B0A0 A          ST      AC0,@AECMF        ; SET CURRENT RTC INTERVAL ERROR FLAG
1227          ;
1227          ;                 ... COMMAND FEEDBACK TEST FOR CROSS STREET ...
1227          ECOM1:
1227 886B A          LD      AC2,AXBUF
1228 8202 A          LD      AC0,D2(AC2)       ; XOCW2 OUTPUT COMMAND
1229 60F2 A          AND     AC0,H3FFF         ; MASK OFF DON'T CARE BITS 15,14
122A 8868 A          LD      AC2,ABBUF
122B 8602 A          LD      AC1,D2(AC2)       ; BCR2 FEEDBACK COMMAND
122C 60F2 A          AND     AC0,H3FFF         ; MASK OFF DON'T CARE BITS 15,14
122D 3482 A          RXOR    AC1,AC0           ; COMPARE, AC1 UNALTERED
122E 110C A          BOC     CCEZ,ECOM2        ; BRANCH IF IDENTICAL (AC0=0)
122F          ;
122F          ;                 ... CROSS STREET COMMAND FEEDBACK ERROR ...
122F 889F A          LD      AC2,AEERR
1230 A606 A          ST      AC1,D6(AC2)       ; STORE BCR2 IN ERROR BUFFER, EERR2
1231 8205 A          LD      AC0,D5(AC2)       ; FETCH ERROR COUNTER AND INCREMENT
1232 4801 A          AISZ    AC0,C1
1233 2100 A          JMP     .+1
1234 A205 A          ST      AC0,D5(AC2)
1235 8200 A          LD      AC0,D0(AC2)
1236 0722 A          SETBIT  D2                ; SET ERROR FLAG BIT 2
1237 A200 A          ST      AC0,D0(AC2)
1238 4C02 A          LI      AC0,C2
```

```
1119 1239 B0A0 A          ST      AC0,@AECMF      ; SET CURRENT RTC INTERVAL ERROR FLAG
1120 123A 2103 A          JMP     ECOMR
1121 123B         ;
1122 123B 90A0 A ECOM2:   LD      AC0,@AECMF      ; TEST CURRENT RTC INTERVAL ERROR FLAG
1123 123C 1501 A          BOC     CCNZ,ECOMR      ; BRANCH IF ERROR
1124 123D         ;
1125 123D 2C3A A          JSR     @AXWDT          ; RESET WATCHDOG TIMER IF NO ERROR
1126 123E         ;
1127 123E 0200 A ECOMR:   RTS     D0              ; RETURN FROM ECOM
1128 123F         ;
```

```
1131 123F          .LIST   LES
1132 123F        ;
1133 123F        ;                 ... LOOP MASTER SYNCHRONIZATION ROUTINE ...
1134 123F        ESYN:
1135 123F 90A2 A        LD      AC0,@AER123     ; FETCH RESET SELECT DATUM
1136 1240 1125 A        BOC     CCEZ,ESYN4A     ; BRANCH IF NO RESET IS HIGH (PULSE)
1137 1241 4D02 A        LI      AC1,C2
1138 1242 3482 A        RXOR    AC1,AC0
1139 1243 1113 A        BOC     CCEZ,ESYN1      ; BRANCH IF ONLY RESET 1 IS HIGH
1140 1244 90A2 A        LD      AC0,@AER123     ; RESTORE RESET SELECT DATUM
1141 1245 4D04 A        LI      AC1,C4
1142 1246 3482 A        RXOR    AC1,AC0
1143 1247 1114 A        BOC     CCEZ,ESYN2      ; BRANCH IF ONLY RESET 2 IS HIGH
1144 1248 90A2 A        LD      AC0,@AER123     ; RESTORE RESET SELECT DATUM
1145 1249 4D08 A        LI      AC1,C8
1146 124A 3482 A        RXOR    AC1,AC0
1147 124B 1115 A        BOC     CCEZ,ESYN3      ; BRANCH IF ONLY RESET 3 IS HIGH
1148 124C 889F A        LD      AC2,AEERR       ; ERROR, MORE THAN ONE RESET IS HIGH
1149 124D 90A2 A        LD      AC0,@AER123     ; RESTORE RESET SELECT DATUM
1150 124E A208 A        ST      AC0,D8(AC2)     ; STORE IN ERROR BUFFER, EERR3
1151 124F 8207 A        LD      AC0,D7(AC2)     ; FETCH ERROR COUNTER AND INCREMENT
1152 1250 4801 A        AISZ    AC0,C1
1153 1251 2100 A        JMP     .+1
1154 1252 A207 A        ST      AC0,D7(AC2)
1155 1253 8200 A        LD      AC0,D0(AC2)
1156 1254 0723 A        SETBIT  D3              ; SET ERROR FLAG BIT 3
1157 1255 A200 A        ST      AC0,D0(AC2)
1158 1256 0200 A        RTS     D0
1159 1257        ;
1160 1257        ESYN1:                          ; RESET 1
1161 1257 2921 A        JSR     ESYA            ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1162 1258 8201 A        LD      AC0,D1(AC2)     ; RESET 1 POSITION (RTC UNITS)
1163 1259 D205 A        SUB     AC0,D5(AC2)     ; COMPARE WITH SYNCH DIAL TIMER
1164 125A 111D A        BOC     CCEZ,ESYN6      ; BRANCH IF DIAL IS AT RESET POINT
1165 125B 210B A        JMP     ESYN4
1166 125C        ;
1167 125C        ESYN2:                          ; RESET 2
1168 125C 291C A        JSR     ESYA            ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1169 125D 8202 A        LD      AC0,D2(AC2)     ; RESET 2 POSITION (RTC UNITS)
1170 125E D205 A        SUB     AC0,D5(AC2)     ; COMPARE WITH SYNCH DIAL TIMER
1171 125F 1118 A        BOC     CCEZ,ESYN6      ; BRANCH IF DIAL IS AT RESET POINT
1172 1260 2106 A        JMP     ESYN4
1173 1261        ;
1174 1261        ESYN3:                          ; RESET 3
1175 1261 2917 A        JSR     ESYA            ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1176 1262 8203 A        LD      AC0,D3(AC2)     ; RESET 3 POSITION (RTC UNITS)
1177 1263 D205 A        SUB     AC0,D5(AC2)     ; COMPARE WITH SYNCH DIAL TIMER
1178 1264 1113 A        BOC     CCEZ,ESYN6      ; BRANCH IF DIAL IS AT RESET POINT
1179 1265 2101 A        JMP     ESYN4
1180 1266        ;
1181 1266 2912 A ESYN4A: JSR     ESYA            ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1182 1267        ESYN4:                          ; INCREMENT SYNCH DIAL TIMER
```

```
1267 7A05 A          ISZ     D5(AC2)
1268 2100 A          JMP     .+1
1269 8200 A          LD      AC0,D0(AC2)      ; FETCH CYCLE LENGTH
126A E205 A          SKG     AC0,D5(AC2)      ; SKIP IF BEFORE END OF CYCLE
126B 2101 A          JMP     .+2
126C 2105 A          JMP     ESYN5
126D 4C00 A          LI      AC0,C0
126E A205 A          ST      AC0,D5(AC2)      ; ZERO SYNCH DIAL TIMER
126F 4C01 A          LI      AC0,C1
1270 B09D A          ST      AC0,@AEPST       ; SET START PULSE
1271 2106 A          JMP     ESYN6
1272         ;
1272         ESYN5:
1272 8204 A          LD      AC0,D4(AC2)      ; FETCH END OF START PULSE
1273 E205 A          SKG     AC0,D5(AC2)      ; SKIP IF BEFORE END OF START PULSE
1274 2101 A          JMP     .+2
1275 2102 A          JMP     ESYN6
1276 4C00 A          LI      AC0,C0
1277 B09D A          ST      AC0,@AEPST       ; RESET START PULSE
1278         ;
1278         ESYN6:
1278 0200 A          RTS     D0               ; RETURN FROM ESYN
1279         ;
1279         ESYA:                            ; COMPUTE BASE ADDRESS OF SYNCH DATA
1279                                          ;    FOR OPERATING DIAL. (AC2)
1279 4C06 A          LI      AC0,C6           ; BUFFER LENGTH = 6
127A 88A1 A          LD      AC2,AESYD        ; BASE ADDRESS FOR DIAL 1
127B 9C83 A          LD      AC3,@ATCDS       ; CURRENT DIAL SELECTION
127C 4BFF A ESYA1:   AISZ    AC3,CM1          ; SKIP IF BASE ADDRESS COMPUTED
127D 2101 A          JMP     .+2
127E 2102 A          JMP     ESYA2
127F 3200 A          RADD    AC0,AC2          ; INCREMENT BY BUFFER LENGTH
1280 21FB A          JMP     ESYA1
1281         ESYA2:
1281 0200 A          RTS     D0               ; RETURN FROM ESYA
1282         ;
```

```
1221 1282                    .LIST   LES
1222 1282          ;
1223 1282          ;                 ... MONITOR INTERCONNECT DATA (R1,R2,R3,D2,D3) ...
1224 1282          EMIC:
1225 1282          ;                 ... NOISE OR TRANSITION FILTER ON D2,D3 ...
1226 1282          ;
1227 1282 9069 A         LD      AC0,@ABDIN      ; ACCESS BDIN
1228 1283 611E A         AND     AC0,H0030       ; EXTRACT D2,D3 TO GET NEW EDNO
1229 1284 8CA5 A         LD      AC3,AEDNO       ; PAST VALUE BUFFER BASE ADDRESS
1230 1285 8700 A         LD      AC1,D0(AC3)     ; UPDATE PAST VALUES
1231 1286 8801 A         LD      AC2,D1(AC3)
1232 1287 A300 A         ST      AC0,D0(AC3)     ; STORE IN BUFFER, EDN0, 0TH PAST VALUE
1233 1288 A701 A         ST      AC1,D1(AC3)     ;                  EDN1, 1ST PAST VALUE
1234 1289 A802 A         ST      AC2,D2(AC3)     ;                  EDN2, 2ND PAST VALUE
1235 128A 3482 A         RXOR    AC1,AC0
1236 128B 1505 A         BOC     CCNZ,EMIC1      ; BRANCH IF EDN0.NE.EDN1
1237 128C 8300 A         LD      AC0,D0(AC3)     ; RESTORE EDN0
1238 128D 3882 A         RXOR    AC2,AC0
1239 128E 1502 A         BOC     CCNZ,EMIC1      ; BRANCH IF EDN0.NE.EDN2
1240 128F 8300 A         LD      AC0,D0(AC3)     ; RESTORE EDN0
1241 1290 B0A4 A         ST      AC0,@AED23      ; THREE PAST VALUES ARE IDENTICAL
1242 1291                                        ;    UPDATE INTERNAL DIAL SELECT DATUM
1243 1291          EMIC1:
1244 1291          ;                 ... NOISE OR TRANSITION FILTER ON R1,R2,R3 ...
1245 1291          ;
1246 1291 9069 A         LD      AC0,@ABDIN      ; ACCESS BDIN
1247 1292 610E A         AND     AC0,H000E       ; EXTRACT R1,R2,R3 TO GET NEW ERN0
1248 1293 8CA3 A         LD      AC3,AERN0       ; PAST VALUE BUFFER BASE ADDRESS
1249 1294 8700 A         LD      AC1,D0(AC3)     ; UPDATE PAST VALUES
1250 1295 8801 A         LD      AC2,D1(AC3)
1251 1296 A300 A         ST      AC0,D0(AC3)     ; STORE IN BUFFER, ERN0, 0TH PAST VALUE
1252 1297 A701 A         ST      AC1,D1(AC3)     ;                  ERN1, 1ST PAST VALUE
1253 1298 A802 A         ST      AC2,D2(AC3)     ;                  ERN2, 2ND PAST VALUE
1254 1299 3482 A         RXOR    AC1,AC0
1255 129A 1505 A         BOC     CCNZ,EMIC2      ; BRANCH IF ERN0.NE.ERN1
1256 129B 8300 A         LD      AC0,D0(AC3)     ; RESTORE ERN0
1257 129C 3882 A         RXOR    AC2,AC0
1258 129D 1502 A         BOC     CCNZ,EMIC2      ; BRANCH IF ERN0.NE.ERN2
1259 129E 8300 A         LD      AC0,D0(AC3)     ; RESTORE ERN0
1260 129F B0A2 A         ST      AC0,@AER123     ; THREE PAST VALUES ARE IDENTICAL
1261 12A0                                        ;    UPDATE INTERNAL RESET SELECT DATUM
1262 12A0          EMIC2:
1263 12A0 0200 A         RTS     D0              ; RETURN FROM EMIC
1264 12A1          ;
1265 12A1          ;                 ... LOCAL DATA ...
1266 12A1 000E A H000E:  .WORD   X'000E          ; BITS 1,2,3 SET
1267 12A2 0030 A H0030:  .WORD   X'0030          ; BITS 4,5 SET
1268 12A3          ;
```

```
1271 12A3                .LIST   LES
1272 12A3        ;
1273 12A3        ;                    ... EXECUTIVE TIMING ROUTINE ...
1274 12A3        ETIM:
1275 12A3 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1276 12A4        ;
1277 12A4 9069 A         LD      AC0,@ABDIN              ; FETCH BDIN
1278 12A5 075E A         SKBIT   D14                     ; SKIP IF WAIT CONDITION
1279 12A6 2101 A         JMP     .+2
1280 12A7 2105 A         JMP     ETIM1
1281 12A8                                                ; BYPASS THESE FNS IN THE WAIT STATE
1282 12A8 909B A         LD      AC0,@AETCY              ; TIMING PLAN CYCLE TIMER
1283 12A9 4801 A         AISZ    AC0,C1                  ; INCREMENT TIMERS
1284 12AA 2100 A         JMP     .+1
1285 12AB B09B A         ST      AC0,@AETCY              ; STORE CYCLE TIMER
1286 12AC B09C A         ST      AC0,@AETST              ; STORE 'WAIT FOR SYNCH START' TIMER
1287 12AD        ;
1288 12AD 90A6 A ETIM1:  LD      AC0,@AEDMIT             ; INCREMENT DETECTOR MONITORING
1289 12AE 4801 A         AISZ    AC0,C1                  ;   INTERVAL TIMER
1290 12AF 2100 A         JMP     .+1
1291 12B0 B0A6 A         ST      AC0,@AEDMIT
1292 12B1        ;
1293 12B1 0200 A ETIMR:  RTS     D0
1294 12B2        ;
```

```
12B2                    .LIST   LES
12B2            ;
12B2            ;                       ... DETECTOR MONITORING ROUTINE ...
12B2            EDET:
12B2 90A6 A             LD      AC0,@AEDMIT     ; FETCH TIMER
12B3 E128 A             SKG     AC0,EDETI       ; SKIP IF DETECTOR MONITORING INTERVAL
12B4                                            ;   HAS BEEN REACHED
12B4 2126 A             JMP     EDETR
12B5            ;
12B5 8C7C A             LD      AC3,AVCVC       ; BASE ADDRESS OF VEHICLE COUNTERS
12B6            ;
12B6 8300 A     EDET1:  LD      AC0,(AC3)       ; FETCH NEXT VEHICLE COUNTER
12B7 E125 A             SKG     AC0,EDETM       ; SKIP IF GT MINIMUM COUNT
12B8 2101 A             JMP     .+2             ; UNDERCOUNTING ERROR
12B9 2119 A             JMP     EDET3           ; NO ERROR
12BA            ;
12BA            ;                       ... UNDERCOUNTING ERROR ...
12BA 3E81 A             RCPY    AC3,AC2         ; COPY CURRENT VEHICLE COUNTER ADDRESS
12BB 847C A             LD      AC1,AVCVC       ; BASE ADDRESS OF VEHICLE COUNTERS
12BC 5101 A             CAI     AC1,C1          ; COMPUTE DETECTOR INDEX
12BD 3600 A             RADD    AC1,AC2
12BE E91F A             SKG     AC2,EDET32      ; SKIP IF INVALID INDEX
12BF 2106 A             JMP     EDET2
12C0            ;
12C0            ;                       ... RECORD DETECTOR INDEX ERROR ...
12C0 3981 A             RCPY    AC2,AC1         ; COPY ERROR DATUM TO AC1
12C1 BCA7 A             ST      AC3,@AEDETT     ; STORE VEHICLE COUNTER ADDRESS IN TEMP
12C2 8C9F A             LD      AC3,AEERR       ; BASE ADDRESS OF ERROR BUFFER
12C3 2C4E A             JSR     @AERE5          ; AC1 HAS ERROR DATUM
12C4 9CA7 A             LD      AC3,@AEDETT     ; RESTORE VEHICLE COUNTER ADDRESS
12C5 210D A             JMP     EDET3
12C6            ;
12C6            ;                       ... RECORD ERROR CURRENT DETECTOR ...
12C6 B8A7 A     EDET2:  ST      AC2,@AEDETT     ; STORE DETECTOR INDEX IN TEMP LOC
12C7 5E01 A             SHL     AC2,D1          ; MPY BY 2 FOR DISP IN ERROR BUFFER
12C8 C8A8 A             ADD     AC2,AEDERR      ; ADD TO BASE ADDRESS
12C9 A201 A             ST      AC0,D1(AC2)     ; STORE UNDERCOUNT VALUE
12CA 8200 A             LD      AC0,D0(AC2)     ; INCREMENT ERROR COUNTER
12CB 4801 A             AISZ    AC0,C1
12CC 2100 A             JMP     .+1
12CD A200 A             ST      AC0,D0(AC2)
12CE            ;
12CE            ;                       ... RECORD GENERAL UNDERCOUNTING ERROR ...
12CE 94A7 A             LD      AC1,@AEDETT     ; TEMP HAS DETECTOR INDEX, ERROR DATUM
12CF BCA7 A             ST      AC3,@AEDETT     ; STORE VEHICLE COUNTER ADDRESS IN TEMP
12D0 8C9F A             LD      AC3,AEERR       ; BASE ADDRESS OF ERROR BUFFER
12D1 2C4D A             JSR     @AERE4          ; AC1 HAS ERROR DATUM
12D2 9CA7 A             LD      AC3,@AEDETT     ; RESTORE VEHICLE COUNTER ADDRESS
12D3            ;
12D3 4C00 A     EDET3:  LI      AC0,C0
12D4 A300 A             ST      AC0,(AC3)       ; ZERO VEHICLE COUNTER
12D5 4801 A             AISZ    AC3,C1          ; INCREMENT VEHICLE COUNTER ADDRESS
```

```
12D6 2100 A          JMP     .+1
12D7 ED07 A          SKG     AC3,EVCVC        ; SKIP IF ALL VEHICLE COUNTERS PROCESSED
12D8 21DD A          JMP     EDET1
12D9 4C00 A          LI      AC0,C0
12DA B0A6 A          ST      AC0,@AEDMIT      ; ZERO DETECTOR MONITORING INT TIMER
12DB         ;
12DB 0200 A EDETR:   RTS     D0               ; RETURN FROM EDET
12DC         ;
12DC         ;                  ... LOCAL DATA ...
12DC 2EE0 A EDETI:   .WORD   12000            ; DET MONITORING INT (0.1 SEC UNITS)
12DD 0001 A EDETM:   .WORD   1                ; MINIMUM VEHICLE COUNT
12DE 001F A EDET32:  .WORD   31               ; MAXIMUM DETECTOR INDEX (0 - 31)
12DF 3134 A EVCVC:   .WORD   VCVC+DETS-1      ; ADDRESS OF LAST DETECTOR VEHICLE CTR
12E0                                          ;    (BASE ADDR + NO. DETECTORS - 1)
12E0         ;
```

```
12E0                .LIST   LES
12E0                ;
12E0                ;                   ... CONSTRUCT STATUS DISPLAY WORD ...
12E0                ESDW:
12E0                ;
12E0                ;                   ... BITS 15 TO 11 ...
12E0 90A2 A         LD      AC0,@AER123     ; R1,R2,R3,IN BITS 1 TO 3
12E1 88A4 A         LD      AC2,AED23
12E2 6A00 A         OR      AC0,(AC2)       ; D2,D3 IN BITS 4,5
12E3 0A80 A         PFLG    SELF            ; ASSURE SEL = 0
12E4 58FA A         ROR     AC0,D6          ; R1,R2,R3,D2,D3 IN BITS 11 TO 15
12E5 6132 A         AND     AC0,HF800       ; ASSURE BITS 10 TO 0 RESET
12E6 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
12E7                ;
12E7                ;                   ... BIT 10 ...
12E7 909D A         LD      AC0,@AEPST
12E8 1103 A         BOC     CCEZ,.+4        ; BRANCH IF NO START PULSE
12E9 3481 A         RCPY    AC1,AC0
12EA 072A A         SETBIT  D10             ; START PULSE
12EB 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
12EC                ;
12EC                ;                   ... BIT 7 ...
12EC 906E A         LD      AC0,@APIP
12ED 1103 A         BOC     CCEZ,.+4        ; BRANCH IF NO PREEMPT IN PROGRESS
12EE 3481 A         RCPY    AC1,AC0
12EF 0727 A         SETBIT  D7              ; PREEMPT IN PROGRESS
12F0 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
12F1                ;
12F1                ;                   ... BIT 6 ...
12F1 907B A         LD      AC0,@AVCBT      ; VEHICLE CLASSIFICATION FLAG
12F2 1103 A         BOC     CCEZ,.+4        ; BRANCH IF NO RECENT BUS CLASS.
12F3 3481 A         RCPY    AC1,AC0
12F4 0726 A         SETBIT  D6              ; BUS CLASSIFIED DURING NEAR PAST
12F5 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
12F6                ;
12F6                ;                   ... BITS 5 THRU 0 REQUIRE MANUAL RESET ...
12F6                ;
12F6                ;                   ... BIT 5 ...
12F6 909F A         LD      AC0,@AEERR
12F7 6121 A         AND     AC0,HFFEF       ; IGNORE DETECTOR UNDERCOUNT ERROR
12F8 1508 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
12F9 906C A         LD      AC0,@AXERR
12FA 1506 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
12FB 906A A         LD      AC0,@ABERR
12FC 1504 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
12FD 9090 A         LD      AC0,@ATERR
12FE 1502 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
12FF 9077 A         LD      AC0,@APERR
1300 1103 A         BOC     CCEZ,.+4
1301                ESDW1:
1301 3481 A         RCPY    AC1,AC0
1302 0725 A         SETBIT  D5              ; SOFTWARE DETECTED ERROR
```

```
1418 1303 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1419 1304          ;
1420 1304          ;                    ... BIT 4 ...
1421 1304 909F A          LD      AC0,@AEERR
1422 1305 6114 A          AND     AC0,H0010        ; EXTRACT DETECTOR UNDERCOUNT FLAG
1423 1306 1103 A          BOC     CCEZ,.+4         ; BRANCH IF NO DETECTOR UNDERCOUNT ERROR
1424 1307 3481 A          RCPY    AC1,AC0
1425 1308 0724 A          SETBIT  D4               ; DETECTOR UNDERCOUNT ERROR
1426 1309 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1427 130A          ;
1428 130A          ;                    ... BIT 2 ...
1429 130A 9069 A          LD      AC0,@ABDIN       ; DISCRETE INPUT WORD
1430 130B 0756 A          SKBIT   D6               ; CMU CONFLICT INDICATION
1431 130C 2103 A          JMP     .+4
1432 130D 3481 A          RCPY    AC1,AC0
1433 130E 0722 A          SETBIT  D2               ; CMU CONFLICT
1434 130F 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1435 1310          ;
1436 1310          ;                    ... BIT 1 ...
1437 1310 9069 A          LD      AC0,@ABDIN       ; DISCRETE INPUT WORD
1438 1311 075F A          SKBIT   D15              ; SKIP IF EXTERNAL DISCONNECT
1439 1312 2103 A          JMP     .+4
1440 1313 3481 A          RCPY    AC1,AC0
1441 1314 0721 A          SETBIT  D1               ; EXTERNAL DISCONNECT
1442 1315 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1443 1316          ;
1444 1316 B46B A          ST      AC1,@AXBUF       ; STORE STATUS DISPLAY WORD
1445 1317          ;
1446 1317 0200 A          RTS     D0               ; RETURN FROM ESDW
1447 1318          ;
1448 1318          ;                    ... LOCAL DATA ...
1449 1318 F800 A HF800:   .WORD   X'F800           ; BITS 10 TO 0 RESET
1450 1319 FFEF A HFFEF:   .WORD   X'FFEF           ; BIT 4 RESET
1451 131A 0010 A H0010:   .WORD   X'0010           ; BIT 4 SET
1452 131B          ;
1453 131B          ;
```

```
1456 131B                    .LIST   LES
1457 131B          ;
1458 131B          ;                    ... END OF CYCLE ROUTINE ...
1459 131B          ;
1460 131B          EEOC:
1461 131B 4C01 A          LI      AC0,C1
1462 131C B086 A          ST      AC0,@ATFST        ; SET 'WAIT FOR SYNCH START' FLAG
1463 131D          ;
1464 131D          ;                    ... SUBSEQUENT FNS MAY RESET THIS FLAG ...
1465 131D          ;
1466 131D          ;
1467 131D          ;                    ... PREEMPT FUNCTIONS ...
1468 131D 2903 A          JSR     EPEOC
1469 131E          ;
1470 131E          ;                    ... TIMING PLAN FUNCTIONS ...
1471 131E 2C37 A          JSR     @ATNCY
1472 131F          ;
1473 131F          ;                    ... EXECUTIVE TIMING FUNCTIONS ...
1474 131F 2913 A          JSR     ETEOC
1475 1320          ;
1476 1320 0200 A          RTS     D0                ; RETURN FROM EEOC
1477 1321          ;
1478 1321          ;                    ... END OF CYCLE PREEMPT FUNCTIONS ...
1479 1321          EPEOC:
1480 1321 9074 A          LD      AC0,@APFREQ       ; TEST FOR PREEMPT MOD OF THE PAST CYCLE
1481 1322 1501 A          BOC     CCNZ,.+2          ; BRANCH IF PREEMPT MOD
1482 1323 2102 A          JMP     .+3
1483 1324 4C00 A          LI      AC0,C0
1484 1325 B086 A          ST      AC0,@ATFST        ; RESET 'WAIT FOR SYNC START' FLAG
1485 1326 4C00 A          LI      AC0,C0
1486 1327 B074 A          ST      AC0,@APFREQ       ; RESET PREEMPT MOD FLAG
1487 1328          ;
1488 1328                                           ; CLEAR PREEMPT MONITOR INDEX FLAGS
1489 1328 8873 A          LD      AC2,APMIF
1490 1329 8200 A          LD      AC0,D0(AC2)
1491 132A 60F0 A          AND     AC0,H7FFF         ; RESET BIT 15
1492 132B A200 A          ST      AC0,D0(AC2)
1493 132C 8201 A          LD      AC0,D1(AC2)
1494 132D 60F0 A          AND     AC0,H7FFF         ; RESET BIT 15
1495 132E A201 A          ST      AC0,D1(AC2)
1496 132F 8202 A          LD      AC0,D2(AC2)
1497 1330 60F0 A          AND     AC0,H7FFF         ; RESET BIT 15
1498 1331 A202 A          ST      AC0,D2(AC2)
1499 1332          ;
1500 1332 0200 A          RTS     D0                ; RETURN FROM EPEOC
1501 1333          ;
1502 1333          ;                    ... END OF CYCLE EXECUTIVE TIMING ROUTINE ...
1503 1333          ETEOC:
1504 1333 8898 A          LD      AC2,AETCY         ; TIMING PLAN CYCLE TIMER
1505 1334 8600 A          LD      AC1,D0(AC2)
1506 1335 E601 A          SKG     AC1,D1(AC2)       ; SKIP IF GREATER THAN PREVIOUS MAXIMUM
1507 1336 2102 A          JMP     ETEOC1
```

```
1508 1337 A601 A          ST      AC1,D1(AC2)      ; STORE NEW MAXIMUM
1509 1338 2104 A          JMP     ETEOC2
1510 1339 3481 A ETEOC1:  RCPY    AC1,AC0
1511 133A D202 A          SUB     AC0,D2(AC2)
1512 133B 1201 A          BOC     CCZP,ETEOC2      ; BRANCH IF GREATER THAN PREVIOUS MIN.
1513 133C A602 A          ST      AC1,D2(AC2)
1514 133D 4000 A ETEOC2:  LI      AC1,C0
1515 133E A600 A          ST      AC1,D0(AC2)      ; ZERO CYCLE TIMER
1516 133F        ;
1517 133F 0200 A          RTS     D0               ; RETURN FROM ETEOC
1518 1340        ;
```

```
1340                .LIST  LER
1340          ;
1340          ;                    ... ERROR RECORDING ROUTINE ...
1340          ;
1340          ;                          UPON ENTRY, AC1 = ERROR DATUM
1340          ;                                      AC3 = BASE ADDR OF ERROR BUFFER
1340          ;                                      AC0 USED TO SET ERROR FLAG
1340          ;                          ENTRY POINT IMPLIES ERROR TYPE
1340          ;
1340          ERE0:
1340 A702 A          ST      AC1,D2(AC3)      ; STORE ERROR DATUM
1341 8301 A          LD      AC0,D1(AC3)      ; FETCH ERROR COUNTER
1342 4801 A          AISZ    AC0,C1           ; INCREMENT
1343 2100 A          JMP     .+1
1344 A301 A          ST      AC0,D1(AC3)
1345 8300 A          LD      AC0,D0(AC3)
1346 0720 A          SETBIT  D0               ; SET ERROR FLAG BIT 0
1347 A300 A          ST      AC0,D0(AC3)
1348 0200 A          RTS     D0               ; RETURN FROM ERE0
1349          ;
1349          ERE1:
1349 A704 A          ST      AC1,D4(AC3)      ; STORE ERROR DATUM
134A 8303 A          LD      AC0,D3(AC3)      ; FETCH ERROR COUNTER
134B 4801 A          AISZ    AC0,C1           ; INCREMENT
134C 2100 A          JMP     .+1
134D A303 A          ST      AC0,D3(AC3)
134E 8300 A          LD      AC0,D0(AC3)
134F 0721 A          SETBIT  D1               ; SET ERROR FLAG BIT 1
1350 A300 A          ST      AC0,D0(AC3)
1351 0200 A          RTS     D0               ; RETURN FROM ERE1
1352          ;
1352          ERE2:
1352 A706 A          ST      AC1,D6(AC3)      ; STORE ERROR DATUM
1353 8305 A          LD      AC0,D5(AC3)      ; FETCH ERROR COUNTER
1354 4801 A          AISZ    AC0,C1           ; INCREMENT
1355 2100 A          JMP     .+1
1356 A305 A          ST      AC0,D5(AC3)
1357 8300 A          LD      AC0,D0(AC3)
1358 0722 A          SETBIT  D2               ; SET ERROR FLAG BIT 2
1359 A300 A          ST      AC0,D0(AC3)
135A 0200 A          RTS     D0               ; RETURN FROM ERE2
135B          ;
135B          ERE3:
135B A708 A          ST      AC1,D8(AC3)      ; STORE ERROR DATUM
135C 8307 A          LD      AC0,D7(AC3)      ; FETCH ERROR COUNTER
135D 4801 A          AISZ    AC0,C1           ; INCREMENT
135E 2100 A          JMP     .+1
135F A307 A          ST      AC0,D7(AC3)
1360 8300 A          LD      AC0,D0(AC3)
1361 0723 A          SETBIT  D3               ; SET ERROR FLAG BIT 3
1362 A300 A          ST      AC0,D0(AC3)
1363 0200 A          RTS     D0               ; RETURN FROM ERE3
```

```
1364          ;
1364          ERE4:
1364 A70A A          ST      AC1,D10(AC3)     ; STORE ERROR DATUM
1365 8309 A          LD      AC0,D9(AC3)      ; FETCH ERROR COUNTER
1366 4801 A          AISZ    AC0,C1           ; INCREMENT
1367 2100 A          JMP     .+1
1368 A309 A          ST      AC0,D9(AC3)
1369 8300 A          LD      AC0,D0(AC3)
136A 0724 A          SETBIT  D4               ; SET ERROR FLAG BIT 4
136B A300 A          ST      AC0,D0(AC3)
136C 0200 A          RTS     D0               ; RETURN FROM ERE4
136D          ;
136D          ERE5:
136D A70C A          ST      AC1,D12(AC3)     ; STORE ERROR DATUM
136E 830B A          LD      AC0,D11(AC3)     ; FETCH ERROR COUNTER
136F 4801 A          AISZ    AC0,C1           ; INCREMENT
1370 2100 A          JMP     .+1
1371 A30B A          ST      AC0,D11(AC3)
1372 8300 A          LD      AC0,D0(AC3)
1373 0725 A          SETBIT  D5               ; SET ERROR FLAG BIT 5
1374 A300 A          ST      AC0,D0(AC3)
1375 0200 A          RTS     D0               ; RETURN FROM ERE5
1376          ;
1376          ERE6:
1376 A70E A          ST      AC1,D14(AC3)     ; STORE ERROR DATUM
1377 830D A          LD      AC0,D13(AC3)     ; FETCH ERROR COUNTER
1378 4801 A          AISZ    AC0,C1           ; INCREMENT
1379 2100 A          JMP     .+1
137A A30D A          ST      AC0,D13(AC3)
137B 8300 A          LD      AC0,D0(AC3)
137C 0726 A          SETBIT  D6               ; SET ERROR FLAG BIT 6
137D A300 A          ST      AC0,D0(AC3)
137E 0200 A          RTS     D0               ; RETURN FROM ERE6
137F          ;
137F          ERE7:
137F A710 A          ST      AC1,D16(AC3)     ; STORE ERROR DATUM
1380 830F A          LD      AC0,D15(AC3)     ; FETCH ERROR COUNTER
1381 4801 A          AISZ    AC0,C1           ; INCREMENT
1382 2100 A          JMP     .+1
1383 A30F A          ST      AC0,D15(AC3)
1384 8300 A          LD      AC0,D0(AC3)
1385 0727 A          SETBIT  D7               ; SET ERROR FLAG BIT 7
1386 A300 A          ST      AC0,D0(AC3)
1387 0200 A          RTS     D0               ; RETURN FROM ERE7
1388          ;
1388          ERE8:
1388 A712 A          ST      AC1,D18(AC3)     ; STORE ERROR DATUM
1389 8311 A          LD      AC0,D17(AC3)     ; FETCH ERROR COUNTER
138A 4801 A          AISZ    AC0,C1           ; INCREMENT
138B 2100 A          JMP     .+1
138C A311 A          ST      AC0,D17(AC3)
138D 8300 A          LD      AC0,D0(AC3)
```

```
1625 138E 0728 A        SETBIT  D8              ; SET ERROR FLAG BIT 8
1626 138F A300 A        ST      AC0,D0(AC3)
1627 1390 0200 A        RTS     D0              ; RETURN FROM ERE8
1628 1391       ;
1629 1391       ERE9:
1630 1391 A714 A        ST      AC1,D20(AC3)    ; STORE ERROR DATUM
1631 1392 8313 A        LD      AC0,D19(AC3)    ; FETCH ERROR COUNTER
1632 1393 4801 A        AISZ    AC0,C1          ; INCREMENT
1633 1394 2100 A        JMP     .+1
1634 1395 A313 A        ST      AC0,D19(AC3)
1635 1396 8300 A        LD      AC0,D0(AC3)
1636 1397 0729 A        SETBIT  D9              ; SET ERROR FLAG BIT 9
1637 1398 A300 A        ST      AC0,D0(AC3)
1638 1399 0200 A        RTS     D0              ; RETURN FROM ERE9
1639 139A       ;
1640 139A       ERE10:
1641 139A A716 A        ST      AC1,D22(AC3)    ; STORE ERROR DATUM
1642 139B 8315 A        LD      AC0,D21(AC3)    ; FETCH ERROR COUNTER
1643 139C 4801 A        AISZ    AC0,C1          ; INCREMENT
1644 139D 2100 A        JMP     .+1
1645 139E A315 A        ST      AC0,D21(AC3)
1646 139F 8300 A        LD      AC0,D0(AC3)
1647 13A0 072A A        SETBIT  D10             ; SET ERROR FLAG BIT 10
1648 13A1 A300 A        ST      AC0,D0(AC3)
1649 13A2 0200 A        RTS     D0              ; RETURN FROM ERE10
1650 13A3       ;
1651 13A3       ERE11:
1652 13A3 A718 A        ST      AC1,D24(AC3)    ; STORE ERROR DATUM
1653 13A4 8317 A        LD      AC0,D23(AC3)    ; FETCH ERROR COUNTER
1654 13A5 4801 A        AISZ    AC0,C1          ; INCREMENT
1655 13A6 2100 A        JMP     .+1
1656 13A7 A317 A        ST      AC0,D23(AC3)
1657 13A8 8300 A        LD      AC0,D0(AC3)
1658 13A9 072B A        SETBIT  D11             ; SET ERROR FLAG BIT 11
1659 13AA A300 A        ST      AC0,D0(AC3)
1660 13AB 0200 A        RTS     D0              ; RETURN FROM ERE11
1661 13AC       ;
1662 13AC       ERE12:
1663 13AC A71A A        ST      AC1,D26(AC3)    ; STORE ERROR DATUM
1664 13AD 8319 A        LD      AC0,D25(AC3)    ; FETCH ERROR COUNTER
1665 13AE 4801 A        AISZ    AC0,C1          ; INCREMENT
1666 13AF 2100 A        JMP     .+1
1667 13B0 A319 A        ST      AC0,D25(AC3)
1668 13B1 8300 A        LD      AC0,D0(AC3)
1669 13B2 072C A        SETBIT  D12             ; SET ERROR FLAG BIT 12
1670 13B3 A300 A        ST      AC0,D0(AC3)
1671 13B4 0200 A        RTS     D0              ; RETURN FROM ERE12
1672 13B5       ;
```

```
1673 13B5                    .PAGE
1674 13B5                    .ASECT
1675 13B5 1400 A             .=X'1400
1676 1400                    ;
```

```
1400                .LIST   LPP
1400          ;
1400          ;                    ... PROGRAMMABLE PANEL ROUTINE ...
1400          ;
1400          ;              (EXECUTED DURING RESTART INITIALIZATION ONLY)
1400          ;
1400          EPAN:
1400          ;
1400          ;                    ... INPUT PROG. PANEL DATA ...
1400 2C3E A         JSR     @ABPAN
1401          ;
1401          ;                    ... CLEAR ERROR FLAGS ...
1401 4C00 A         LI      AC0,C0
1402 B0CD A         ST      AC0,@AEP1E        ; PLAN 1
1403 B0CE A         ST      AC0,@AEP2E        ; PLAN 2
1404 B0CF A         ST      AC0,@AEP3E        ; PLAN 3
1405 B0D0 A         ST      AC0,@AEPDVE       ; DATA VALIDITY
1406          ;
1406          ;                    ... TURN OFF ERROR LIGHTS ...
1406 8CEA A         LD      AC3,APP1U         ; PERIPHERAL DEVICE ADDRESS PLAN 1
1407 2C57 A         JSR     @AXLOF1
1408          ;
1408 8CEB A         LD      AC3,APP2U         ; PERIPHERAL DEVICE ADDRESS PLAN 2
1409 2C57 A         JSR     @AXLOF1
140A          ;
140A 8CEC A         LD      AC3,APP3U         ; PERIPHERAL DEVICE ADDRESS PLAN 3
140B 2C57 A         JSR     @AXLOF1
140C          ;
140C 8CE9 A         LD      AC3,APPDU         ; DEVICE ADDRESS DATA VALIDITY LIGHT
140D 2C59 A         JSR     @AXLOF2
140E          ;
140E          ;                    ... COMPUTE SUM OF FIXED INTERVALS IN PLAN 1 ...
140E 888B A         LD      AC2,ATPM1
140F 2951 A         JSR     EPFI
1410 8CC5 A         LD      AC3,AEPFT
1411 A700 A         ST      AC1,D0(AC3)       ; STORE SUM
1412          ;
1412          ;                    ... COMPUTE SUM OF FIXED INTERVALS IN PLAN 2 ...
1412 888C A         LD      AC2,ATPM2
1413 294D A         JSR     EPFI
1414 8CC5 A         LD      AC3,AEPFT
1415 A701 A         ST      AC1,D1(AC3)       ; STORE SUM
1416          ;
1416          ;                    ... COMPUTE SUM OF FIXED INTERVALS IN PLAN 3 ...
1416 888D A         LD      AC2,ATPM3
1417 2949 A         JSR     EPFI
1418 8CC5 A         LD      AC3,AEPFT
1419 A702 A         ST      AC1,D2(AC3)       ; STORE SUM
141A          ;
141A          ;                    ... TEST ADJUSTABLE INTERVALS FOR 0 DURATION ...
141A 888B A         LD      AC2,ATPM1
141B 9493 A         LD      AC1,@ATAI1        ; RELATIVE INDEX OF ADJ INT 1
```

```
141C 3600 A        RADD    AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 1
141D 8200 A        LD      AC0,D0(AC2)      ; FETCH DURATION
141E 1107 A        BOC     CCEZ,EPAN1       ; BRANCH IF ZERO, NO ERROR
141F 8CD1 A        LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
1420          ;
1420          ;                 ... RECORD ERROR IN MASTER TIMING PLAN ...
1420          EPANE:
1420 3981 A        RCPY    AC2,AC1          ; SAVE ADDRESS AS ERROR DATUM
1421 2C49 A        JSR     @AEREO
1422 2C3A A EPANE1: JSR    @AXWDT           ; ISSUE RESET PULSE TO WATCHDOG TIMER
1423          ;
1423 2C2F A        JSR     @AEADC           ; FATAL ERROR, DISCONNECT, NO RETURN
1424          ;
1424 2100 A        JMP     .+1              ; THIS INST SHOULD NEVER BE REACHED
1425 21FE A        JMP     .-1
1426          ;
1426          EPAN1:
1426 888B A        LD      AC2,ATPM1
1427 9494 A        LD      AC1,@ATAI2       ; RELATIVE INDEX OF ADJ INT 2
1428 3600 A        RADD    AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 2
1429 8200 A        LD      AC0,D0(AC2)      ; FETCH DURATION
142A 1102 A        BOC     CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
142B 8CD1 A        LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
142C 21F3 A        JMP     EPANE
142D          ;
142D 888C A        LD      AC2,ATPM2
142E 9493 A        LD      AC1,@ATAI1       ; RELATIVE INDEX OF ADJ INT 1
142F 3600 A        RADD    AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 1
1430 8200 A        LD      AC0,D0(AC2)      ; FETCH DURATION
1431 1102 A        BOC     CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1432 8CD2 A        LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1433 21EC A        JMP     EPANE
1434          ;
1434 888C A        LD      AC2,ATPM2
1435 9494 A        LD      AC1,@ATAI2       ; RELATIVE INDEX OF ADJ INT 2
1436 3600 A        RADD    AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 2
1437 8200 A        LD      AC0,D0(AC2)      ; FETCH DURATION
1438 1102 A        BOC     CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1439 8CD2 A        LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
143A 21E5 A        JMP     EPANE
143B          ;
143B 888D A        LD      AC2,ATPM3
143C 9493 A        LD      AC1,@ATAI1       ; RELATIVE INDEX OF ADJ INT 1
143D 3600 A        RADD    AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 1
143E 8200 A        LD      AC0,D0(AC2)      ; FETCH DURATION
143F 1102 A        BOC     CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1440 8CD3 A        LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
1441 21DE A        JMP     EPANE
1442          ;
1442 888D A        LD      AC2,ATPM3
1443 9494 A        LD      AC1,@ATAI2       ; RELATIVE INDEX OF ADJ INT 2
1444 3600 A        RADD    AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 2
```

```
1783 1445 8200 A        LD      AC0,D0(AC2)     ; FETCH DURATION
1784 1446 1102 A        BOC     CCEZ,.+3        ; BRANCH IF ZERO, NO ERROR
1785 1447 8CD3 A        LD      AC3,AEPEF3      ; BASE ADDRESS OF ERROR BUFFER
1786 1448 21D7 A        JMP     EPANE
1787 1449       ;
1788 1449       ;                   ... PROCESS PLAN 1 DATA ...
1789 1449 2C41 A        JSR     @AEPP1
1790 144A       ;
1791 144A       ;                   ... PROCESS PLAN 2 DATA ...
1792 144A 2C42 A        JSR     @AEPP2
1793 144B       ;
1794 144B       ;                   ... PROCESS PLAN 3 DATA ...
1795 144B 2C43 A        JSR     @AEPP3
1796 144C       ;
1797 144C       ;                   ... TEST ERROR FLAGS ...
1798 144C 90CD A        LD      AC0,@AEP1E      ; PLAN 1
1799 144D 1102 A        BOC     CCEZ,.+3        ; BRANCH IF NO ERROR
1800 144E 8CEA A        LD      AC3,APP1U
1801 144F 2C56 A        JSR     @AXLON1         ; TURN ON ERROR LIGHT
1802 1450       ;
1803 1450 90CE A        LD      AC0,@AEP2E      ; PLAN 2
1804 1451 1102 A        BOC     CCEZ,.+3        ; BRANCH IF NO ERROR
1805 1452 8CEB A        LD      AC3,APP2U
1806 1453 2C56 A        JSR     @AXLON1         ; TURN ON ERROR LIGHT
1807 1454       ;
1808 1454 90CF A        LD      AC0,@AEP3E      ; PLAN 3
1809 1455 1102 A        BOC     CCEZ,.+3        ; BRANCH IF NO ERROR
1810 1456 8CEC A        LD      AC3,APP3U
1811 1457 2C56 A        JSR     @AXLON1         ; TURN ON ERROR LIGHT
1812 1458       ;
1813 1458 90D0 A        LD      AC0,@AEPDVE     ; DATA VALIDITY
1814 1459 1103 A        BOC     CCEZ,.+4        ; BRANCH IF NO ERROR
1815 145A 8CE9 A        LD      AC3,APPDU
1816 145B 2C58 A        JSR     @AXLON2         ; TURN ON ERROR LIGHT
1817 145C 21C5 A        JMP     EPANE1          ; DISCONNECT
1818 145D       ;
1819 145D       ;                   ... UPDATE DATA BASE ...
1820 145D 2C44 A        JSR     @AEPUP
1821 145E       ;
1822 145E 8CE9 A        LD      AC3,APPDU
1823 145F 2C57 A        JSR     @AXLOF1         ; ASSURE PENDING INPUT LIGHT IS OFF
1824 1460       ;
1825 1460 0200 A        RTS     D0              ; RETURN FROM EPAN
1826 1461       ;
1827 1461       ;                   ... COMPUTE SUM OF INTERVALS IN MASTER PLAN ...
1828 1461
1829 1461       ;                           AC2 = BASE ADDRESS OF MASTER TIMING PLAN
1830 1461       ;
1831 1461       ;                           AC1 RETURNS SUM OF INTERVAL DURATIONS
1832 1461       ;
1833 1461       EPFI:
1834 1461 4D00 A        LI      AC1,C0          ; CLEAR SUM
```

```
1835 1462 8200 A EPFI1:  LD     AC0,D0(AC2)     ; FETCH INTERVAL DURATION
1836 1463 1102 A         BOC    CCEZ,.+3        ; BRANCH IF ZERO
1837 1464 1B04 A         BOC    CCZN,EPFIR      ; BRANCH IF NEGATIVE, END OF PLAN
1838 1465 3100 A         RADD   AC0,AC1         ; ADD DURATION TO SUM
1839 1466 4A01 A         AISZ   AC2,C1          ; INCREMENT ADDRESS
1840 1467 2100 A         JMP    .+1
1841 1468 21F9 A         JMP    EPFI1           ; CONTINUE LOOP
1842 1469        ;
1843 1469 0200 A EPFIR:  RTS    D0              ; RETURN FROM EPFI
1844 146A        ;
1845 146A        ;                    ... LOCAL DATA ...
1846 146A 00C8 A EPCY:   .WORD  200             ; MAXIMUM CYCLE LENGTH (SECONDS)
1847 146B 00C8 A EPIN1:  .WORD  200             ; LIMIT ON MAXIMUM EXTENSION (SECONDS)
1848 146C 000A A E10:    .WORD  10              ; CONSTANT FOR DIVISION
1849 146D 0064 A E100:   .WORD  100             ; PERCENTAGE CONVERSION CONSTANT
1850 146E        ;
```

```
1853 146E                .LIST   LPP
1854 146E        ;
1855 146E        ;                  ... PROG PANEL MONITOR ROUTINE ...
1856 146E        ;
1857 146E        ;                      (EXECUTED DURING RTC SEQUENCE)
1858 146E        ;
1859 146E      EPPM:
1860 146E 90BB A          LD      AC0,@ABPPD      ; FETCH PROG PANEL DISCRETES
1861 146F 6153 A          AND     AC0,H0400       ; EXTRACT INPUT REQUEST SWITCH STATE
1862 1470 1502 A          BOC     CCNZ,.+3        ; BRANCH IF INPUT REQUEST ON
1863 1471        ;
1864 1471 B0C6 A          ST      AC0,@AEPUF      ; (AC0) = 0, ASSURE UPDATE FLAG RESET
1865 1472 2105 A          JMP     EPPM1
1866 1473        ;
1867 1473 90C8 A          LD      AC0,@AEPIR      ; FETCH PREVIOUS STATE OF INPUT REQ
1868 1474 1503 A          BOC     CCNZ,.+4        ; BRANCH IF SWITCH WAS ON
1869 1475        ;
1870 1475 4CFF A          LI      AC0,CM1
1871 1476 B0C7 A          ST      AC0,@AEPCF      ; SET FLAG FOR INPUT OF THUMBWHEELS
1872 1477 814B A          LD      AC0,H0400       ; RESTORE SWITCH DATUM
1873 1478        ;
1874 1478 B0C8 A EPPM1:   ST      AC0,@AEPIR      ; SAVE STATE OF INPUT REQUEST
1875 1479        ;
1876 1479 90C7 A          LD      AC0,@AEPCF
1877 147A 1147 A          BOC     CCEZ,EPPMR      ; BRANCH IF NOT INPUT PROCESSING SEQ
1878 147B        ;
1879 147B 88C7 A          LD      AC2,AEPCF       ; ADDRESS OF COMPUTATION FLAG
1880 147C        ;
1881 147C 4DFF A          LI      AC1,CM1
1882 147D F600 A          SKNE    AC1,(AC2)       ; SKIP IF NOT THUMBWHEEL INPUT
1883 147E 2101 A          JMP     .+2
1884 147F 210B A          JMP     EPPM2
1885 1480        ;
1886 1480        ;                  ... CLEAR ERROR FLAGS ...
1887 1480 4C00 A          LI      AC0,C0
1888 1481 B0CD A          ST      AC0,@AEP1E      ; PLAN 1
1889 1482 B0CE A          ST      AC0,@AEP2E      ; PLAN 2
1890 1483 B0CF A          ST      AC0,@AEP3E      ; PLAN 3
1891 1484 B0D0 A          ST      AC0,@AEPDVE     ; DATA VALIDITY
1892 1485        ;
1893 1485        ;                  ... TURN OFF ERROR LIGHT ...
1894 1485        ;
1895 1485 8CE9 A          LD      AC3,APPDU       ; DEVICE ADDRESS DATA VALIDITY LIGHT
1896 1486 2C59 A          JSR     @AXLOF2
1897 1487        ;
1898 1487 2C3E A          JSR     @ABPAN          ; INPUT THUMBWHEEL DATA
1899 1488        ;
1900 1488 4D01 A          LI      AC1,C1
1901 1489 B4C7 A          ST      AC1,@AEPCF      ; SET FLAG FOR PLAN 1 PROCESSING
1902 148A 2137 A          JMP     EPPMR
1903 148B        ;
1904 148B 4D01 A EPPM2:   LI      AC1,C1
```

```
148C F600 A          SKNE   AC1,(AC2)        ; SKIP IF NOT PLAN 1 PROCESSING
148D 2101 A          JMP    .+2
148E 210A A          JMP    EPPM3
148F          ;
149F          ;                  ... TURN OFF ERROR LIGHT ...
148F 8CEA A          LD     AC3,APP1U        ; PERIPHERAL DEVICE ADDRESS PLAN 1
1490 2C57 A          JSR    @AXLOF1
1491          ;
1491 2C41 A          JSR    @AEPP1           ; PLAN 1 PROCESSING
1492          ;
1492          ;                  ... TEST ERROR FLAG ...
1492 90CD A          LD     AC0,@AEP1E       ; PLAN 1
1493 1102 A          BOC    CCEZ,.+3         ; BRANCH IF NO ERROR
1494 8CEA A          LD     AC3,APP1U
1495 2C56 A          JSR    @AXLON1          ; TURN ON ERROR LIGHT
1496          ;
1496 4D02 A          LI     AC1,C2
1497 B4C7 A          ST     AC1,@AEPCF       ; SET FLAG FOR PLAN 2 PROCESSING
1498 2129 A          JMP    EPPMR
1499          ;
1499 4D02 A EPPM3:   LI     AC1,C2
149A F600 A          SKNE   AC1,(AC2)        ; SKIP IF NOT PLAN 2 PROCESSING
149B 2101 A          JMP    .+2
149C 210A A          JMP    EPPM4
149D          ;
149D          ;                  ... TURN OFF ERROR LIGHT ...
149D          ;
149D 8CEB A          LD     AC3,APP2U        ; PERIPHERAL DEVICE ADDRESS PLAN 2
149E 2C57 A          JSR    @AXLOF1
149F          ;
149F 2C42 A          JSR    @AEPP2           ; PLAN 2 PROCESSING
14A0          ;
14A0          ;                  ... TEST ERROR FLAG ...
14A0 90CE A          LD     AC0,@AEP2E       ; PLAN 2
14A1 1102 A          BOC    CCEZ,.+3         ; BRANCH IF NO ERROR
14A2 8CEB A          LD     AC3,APP2U
14A3 2C56 A          JSR    @AXLON1          ; TURN ON ERROR LIGHT
14A4          ;
14A4 4D03 A          LI     AC1,C3
14A5 B4C7 A          ST     AC1,@AEPCF       ; SET FLAG FOR PLAN 3 PROCESSING
14A6 211B A          JMP    EPPMR
14A7          ;
14A7 4D03 A EPPM4:   LI     AC1,C3
14A8 F600 A          SKNE   AC1,(AC2)        ; SKIP IF NOT PLAN 3 PROCESSING
14A9 2101 A          JMP    .+2
14AA 210A A          JMP    EPPM5
14AB          ;
14AB          ;                  ... TURN OFF ERROR LIGHT ...
14AB 8CEC A          LD     AC3,APP3U        ; PERIPHERAL DEVICE ADDRESS PLAN 3
14AC 2C57 A          JSR    @AXLOF1
14AD          ;
14AD 2C43 A          JSR    @AEPP3           ; PLAN 3 PROCESSING
```

```
14AE          ;
14AE          ;                    ... TEST ERROR FLAG ...
14AE 90CF A          LD      AC0,@AEP3E       ; PLAN 3
14AF 1102 A          BOC     CCEZ,.+3         ; BRANCH IF NO ERROR
14B0 8CEC A          LD      AC3,APP3U
14B1 2C56 A          JSR     @AXLON1          ; TURN ON ERROR LIGHT
14B2          ;
14B2 4D04 A          LI      AC1,C4
14B3 B4C7 A          ST      AC1,@AEPCF       ; SET FLAG FOR COMPLETION OF INPUT PROC
14B4 210D A          JMP     EPPMR
14B5          ;
14B5 4D04 A EPPM5:   LI      AC1,C4
14B6 F600 A          SKNE    AC1,(ACZ)        ; SKIP IF NOT COMPLETION OF INPUT PROC
14B7 2101 A          JMP     .+2
14B8 2109 A          JMP     EPPMR
14B9          ;
14B9          ;                    ... TEST ERROR FLAG ...
14B9 90D0 A          LD      AC0,@AEPDVE      ; DATA VALIDITY
14BA 1103 A          BOC     CCEZ,.+4         ; BRANCH IF NO ERROR
14BB 8CE9 A          LD      AC3,APPDU
14BC 2C58 A          JSR     @AXLON2          ; TURN ON ERROR LIGHT
14BD 2102 A          JMP     .+3              ; DATA VALIDITY ERROR, DO NOT UPDATE
14BE          ;
14BE 4D01 A          LI      AC1,C1
14BF B4C6 A          ST      AC1,@AEPUF       ; SET FLAG FOR DATA BASE UPDATE ENABLE
14C0          ;
14C0 4D00 A          LI      AC1,C0
14C1 B4C7 A          ST      AC1,@AEPCF       ; RESET COMPUTE FLAG
14C2          ;
14C2 0200 A EPPMR:   RTS     D0               ; RETURN FROM EPPM
14C3          ;
14C3          ;                    ... LOCAL DATA ...
14C3 0400 A H0400:   .WORD   X'0400           ; BIT 10 SET
14C4          ;
14C4          ;
```

```
14C4                .LIST   LPP
14C4          ;
14C4          ;                 ... PROCESS PROG. PANEL DATA FOR PLAN 1 ...
14C4          EPP1:
14C4          ;
14C4          ;                 ... PROCESS CYCLE LENGTH ...
14C4 88BC A         LD      AC2,ABPD1
14C5 8200 A         LD      AC0,D0(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
14C6 2C46 A         JSR     @AEPBI3         ; CONVERTED VALUE RETURNED IN AC1
14C7 2128 A         JMP     EPP10           ; DATA VALIDITY ERROR
14C8 8CBF A         LD      AC3,AEPD1
14C9 A700 A         ST      AC1,D0(AC3)     ; SAVE CONVERTED INTEGER VALUE (SECONDS)
14CA 88C9 A         LD      AC2,AEPCY       ; MAXIMUM CYCLE LENGTH TEST (SECONDS)
14CB E600 A         SKG     AC1,D0(AC2)     ; SKIP IF .GT. MAXIMUM
14CC 2107 A         JMP     .+8
14CD          ;
14CD          ;                 ... RECORD CYCLE LENGTH INPUT ERROR ...
14CD 8CD1 A         LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
14CE 2C51 A         JSR     @AERE8          ; AC1 = ERROR DATUM
14CF 4C01 A         LI      AC0,C1
14D0 B0CD A         ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
14D1 94C9 A         LD      AC1,@AEPCY      ; SET CYCLE LENGTH TO MAXIMUM (SECONDS)
14D2 8CBF A         LD      AC3,AEPD1
14D3 A700 A         ST      AC1,D0(AC3)     ; SAVE MAXIMUM CYCLE LENGTH (SECONDS)
14D4          ;
14D4 2C47 A         JSR     @AEM10          ; CONVERT TO 0.1 SECOND UNITS IN AC1
14D5 8CC2 A         LD      AC3,AEPCD1
14D6 A703 A         ST      AC1,D3(AC3)     ; SAVE CYCLE LENGTH (0.1 SEC)
14D7          ;
14D7 88C5 A         LD      AC2,AEPFT       ; MINIMUM CYCLE LENGTH TEST (0.1 SEC)
14D8 8200 A         LD      AC0,D0(AC2)     ; SUM OF FIXED INTERVALS
14D9 5001 A         CAI     AC0,C1          ; NEGATE
14DA 3400 A         RADD    AC1,AC0         ; AC0 = ADJ. TIME IF NON-NEG.
14DB 1B02 A         BOC     CCZN,.+3        ; BRANCH IF CYCLE .LE. MINIMUM
14DC A301 A         ST      AC0,D1(AC3)     ; SAVE ADJUSTABLE TIME IN ADJ. INT. 2
14DD 2109 A         JMP     .+10
14DE          ;
14DE          ;                 ... CYCLE LENGTH INPUT ERROR (TOO SHORT) ...
14DE 4C00 A         LI      AC0,C0          ; NO ADJUSTABLE TIME
14DF A301 A         ST      AC0,D1(AC3)     ; SAVE 0 IN ADJUSTABLE INTERVAL 2
14E0          ;
14E0 8200 A         LD      AC0,D0(AC2)     ; RESTORE MIN. SUM OF FIXED INTERVALS
14E1 8CC2 A         LD      AC3,AEPCD1
14E2 A303 A         ST      AC0,D3(AC3)     ; STORE MIN CYCLE (0.1 SEC)
14E3 8CD1 A         LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
14E4 2C51 A         JSR     @AERE8          ; AC1 = ERROR DATUM
14E5 4C01 A         LI      AC0,C1
14E6 B0CD A         ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
14E7          ;
14E7          ;                 ... COMPUTE MAX DWELL TIME FOR SYNCH (24% OF CYCLE) ...
14E7 8CBF A         LD      AC3,AEPD1
14E8 8700 A         LD      AC1,D0(AC3)     ; CYCLE LENGTH (SECONDS)
```

```
14F9 2C48 A          JSR    @AEM24          ; MULTIPLY BY 24
14EA 4C00 A          LI     AC0,C0          ; PREPARE FOR DIVIDE
14EB 88CC A          LD     AC2,AE10        ; *10/100 TO GET 0.1 SEC UNITS
14EC 0690 A          DIV    D0(AC2)
14ED 0000 A
14EE 8CC2 A          LD     AC3,AEPCD1
14EF A708 A          ST     AC1,D8(AC3)     ; MAX DWELL 24% OF CYCLE (0.1 SEC)
14F0        ;
14F0        ;                  ... COMPUTE START PULSE DURATION (3% OF CYCLE) ...
14F0 5DFD A          SHR    AC1,D3          ; DIVIDE BY 8
14F1 A707 A          ST     AC1,D7(AC3)     ; START PULSE DURATION 3% OF CYCLE
14F2 2105 A          JMP    .+6
14F3        ;
14F3        ;                  ... RECORD CYCLE LENGTH VALIDITY ERROR ...
14F3        EPP10:
14F3 8CD1 A          LD     AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
14F4 2C4A A          JSR    @AERE1          ; AC1 = ERROR DATUM
14F5 4C01 A          LI     AC0,C1
14F6 B0CD A          ST     AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
14F7 B0D0 A          ST     AC0,@AEPDVE     ; SET ERROR FLAG FOR DATA VALIDITY
14F8        ;
14F8        ;                  ... PROCESS OFFSET 1 ...
14F8 88BC A          LD     AC2,ABPD1
14F9 8201 A          LD     AC0,D1(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
14FA 2C46 A          JSR    @AEPBI3         ; CONVERT DATUM TO INTEGER IN AC1
14FB 2110 A          JMP    EPP11           ; DATA VALIDITY ERROR
14FC 8CBF A          LD     AC3,AEPD1
14FD A701 A          ST     AC1,D1(AC3)     ; SAVE CONVERTED VALUE (SECONDS)
14FE E700 A          SKG    AC1,D0(AC3)     ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
14FF 2106 A          JMP    .+7
1500        ;
1500        ;                  ... RECORD OFFSET INPUT ERROR ...
1500 8CD1 A          LD     AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
1501 2C52 A          JSR    @AERE9          ; AC1 = ERROR DATUM
1502 4C01 A          LI     AC0,C1
1503 B0CD A          ST     AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
1504 4DF6 A          LI     AC1,CM10        ; SET OUT OF RANGE RESET TO -1 SECOND
1505 2103 A          JMP    .+4
1506        ;
1506        ;                  ... COMPUTE RESET ...
1506 D700 A          SUB    AC1,D0(AC3)     ; OFFSET - CYCLE
1507 5101 A          CAI    AC1,C1          ; NEGATE
1508 2C47 A          JSR    @AEM10          ; MULTIPLY AC1 BY 10
1509 8CC2 A          LD     AC3,AEPCD1
150A A704 A          ST     AC1,D4(AC3)     ; SAVE RESET (0.1 SECONDS)
150B 2105 A          JMP    .+6
150C        ;
150C        ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
150C 8CD1 A EPP11:   LD     AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
150D 2C4B A          JSR    @AERE2          ; AC1 = ERROR DATUM
150E 4C01 A          LI     AC0,C1
150F B0CD A          ST     AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
```

```
2097 1510 B0D0 A        ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
2098 1511        ;
2099 1511        ;                  ... PROCESS OFFSET 2 ...
2100 1511 88BC A        LD      AC2,ABPD1
2101 1512 8202 A        LD      AC0,D2(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2102 1513 2C46 A        JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
2103 1514 2110 A        JMP     EPP12            ; DATA VALIDITY ERROR
2104 1515 8CBF A        LD      AC3,AEPD1
2105 1516 A702 A        ST      AC1,D2(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
2106 1517 E700 A        SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
2107 1518 2106 A        JMP     .+7
2108 1519        ;
2109 1519        ;                  ... RECORD OFFSET INPUT ERROR ...
2110 1519 8CD1 A        LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2111 151A 2C53 A        JSR     @AERE10          ; AC1 = ERROR DATUM
2112 151B 4C01 A        LI      AC0,C1
2113 151C B0CD A        ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2114 151D 4DF6 A        LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
2115 151E 2103 A        JMP     .+4
2116 151F        ;
2117 151F        ;                  ... COMPUTE RESET ...
2118 151F D700 A        SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
2119 1520 5101 A        CAI     AC1,C1           ; NEGATE
2120 1521 2C47 A        JSR     @AEM10           ; MULTIPLY AC1 BY 10
2121 1522 8CC2 A        LD      AC3,AEPCD1
2122 1523 A705 A        ST      AC1,D5(AC3)      ; SAVE RESET (0.1 SECONDS)
2123 1524 2105 A        JMP     .+6
2124 1525        ;
2125 1525        ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
2126 1525 8CD1 A EPP12: LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2127 1526 2C4C A        JSR     @AERE3           ; AC1 = ERROR DATUM
2128 1527 4C01 A        LI      AC0,C1
2129 1528 B0CD A        ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2130 1529 B0D0 A        ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
2131 152A        ;
2132 152A        ;                  ... PROCESS OFFSET 3 ...
2133 152A 88BC A        LD      AC2,ABPD1
2134 152B 8203 A        LD      AC0,D3(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2135 152C 2C46 A        JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
2136 152D 2110 A        JMP     EPP13            ; DATA VALIDITY ERROR
2137 152E 8CBF A        LD      AC3,AEPD1
2138 152F A703 A        ST      AC1,D3(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
2139 1530 E700 A        SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
2140 1531 2106 A        JMP     .+7
2141 1532        ;
2142 1532        ;                  ... RECORD OFFSET INPUT ERROR ...
2143 1532 8CD1 A        LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2144 1533 2C54 A        JSR     @AERE11          ; AC1 = ERROR DATUM
2145 1534 4C01 A        LI      AC0,C1
2146 1535 B0CD A        ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2147 1536 4DF6 A        LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
2148 1537 2103 A        JMP     .+4
```

```
1538          ;
1538          ;                    ... COMPUTE RESET ...
1538 D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
1539 5101 A          CAI     AC1,C1           ; NEGATE
153A 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
153B 8CC2 A          LD      AC3,AEPCD1
153C A706 A          ST      AC1,D6(AC3)      ; SAVE RESET (0.1 SECONDS)
153D 2105 A          JMP     .+6
153E          ;
153E          ;                    ... RECORD OFFSET DATA VALIDITY ERROR ...
153E 8CD1 A EPP13:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
153F 2C4D A          JSR     @AERE4           ; AC1 = ERROR DATUM
1540 4C01 A          LI      AC0,C1
1541 B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
1542 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1543          ;
1543          ;                    ... PROCESS MSG SPLIT % ...
1543 88BC A          LD      AC2,ABPD1
1544 8204 A          LD      AC0,D4(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1545 2C45 A          JSR     @AEPBI2          ; CONVERT DATUM TO INTEGER IN AC1
1546 2114 A          JMP     EPP14            ; DATA VALIDITY ERROR
1547 8CBF A          LD      AC3,AEPD1
1548 A704 A          ST      AC1,D4(AC3)      ; SAVE CONVERTED VALUE (%)
1549          ;
1549 88C2 A          LD      AC2,AEPCD1
154A 8201 A          LD      AC0,D1(AC2)      ; FETCH ADJUSTABLE TIME
154B 1B01 A          BOC     CCZN,.+2         ; TEST FOR POSITIVE
154C 2102 A          JMP     .+3
154D          ;
154D 4D00 A          LI      AC1,C0           ; STORE 0 FOR BOTH ADJ. INTS.
154E 2106 A          JMP     .+7
154F          ;
154F 4C00 A          LI      AC0,C0           ; PREPARE FOR MULTIPLY
1550 0680 A          MPY     D1(AC2)          ; MSG(%) * ADJ. TIME (0.1 SEC)
1551 0001 A
1552 88CB A          LD      AC2,AE100        ; DIVIDE BY 100
1553 0690 A          DIV     D0(AC2)
1554 0000 A
1555          ;
1555 88C2 A          LD      AC2,AEPCD1
1556 A600 A          ST      AC1,D0(AC2)      ; STORE ADJ. INTERVAL 1
1557 5101 A          CAI     AC1,C1           ; SUBTRACT ADJ. INT. 1 FROM TOTAL
1558 C601 A          ADD     AC1,D1(AC2)
1559 A601 A          ST      AC1,D1(AC2)      ; STORE ADJ. INT. 2
155A 2105 A          JMP     .+6
155B          ;
155B          ;                    ... RECORD MSG DATA VALIDITY ERROR ...
155B 8CD1 A EPP14:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
155C 2C4E A          JSR     @AERE5           ; AC1 = ERROR DATUM
155D 4C01 A          LI      AC0,C1
155E B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
155F B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
```

```
1560          ;
1560          ;                    ... PROCESS INT1 ...
1560 88BC A          LD      AC2,ABPD1
1561 8205 A          LD      AC0,D5(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1562 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
1563 210E A          JMP     EPP15            ; DATA VALIDITY ERROR
1564 8CBF A          LD      AC3,AEPD1
1565 A705 A          ST      AC1,D5(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
1566 88CA A          LD      AC2,AEPIN1
1567 E600 A          SKG     AC1,D0(AC2)      ; SKIP IF INT1 .GT. LIMIT
1568 2105 A          JMP     .+6
1569          ;
1569          ;                    ... RECORD INT1 INPUT ERROR ...
1569 8CD1 A          LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
156A 2C55 A          JSR     @AERE12          ; AC1 = ERROR DATUM
156B 4C01 A          LI      AC0,C1
156C B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
156D 94CA A          LD      AC1,@AEPIN1      ; SET MAXIMUM EXTENSION TO LIMIT (SEC)
156E          ;
156E 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
156F 8CC2 A          LD      AC3,AEPCD1
1570 A702 A          ST      AC1,D2(AC3)      ; STORE MAXIMUM EXTENSION (0.1 SEC)
1571 2105 A          JMP     .+6
1572          ;
1572          ;                    ... RECORD INT1 DATA VALIDITY ERROR ...
1572 8CD1 A EPP15:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
1573 2C4F A          JSR     @AERE6           ; AC1 = ERROR DATUM
1574 4C01 A          LI      AC0,C1
1575 B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
1576 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1577          ;
1577 0200 A          RTS     D0               ; RETURN FROM EPP1
1578          ;
```

```
1578                .LIST   LPP
1578        ;
1578        ;                   ... PROCESS PROG. PANEL DATA FOR PLAN 2 ...
1578        EPP2:
1578        ;
1578        ;                   ... PROCESS CYCLE LENGTH ...
1578 88BD A         LD      AC2,A8PD2
1579 8200 A         LD      AC0,D0(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
157A 2C46 A         JSR     @AEPBI3         ; CONVERTED VALUE RETURNED IN AC1
157B 212B A         JMP     EPP20           ; DATA VALIDITY ERROR
157C 8CC0 A         LD      AC3,AEPD2
157D A700 A         ST      AC1,D0(AC3)     ; SAVE CONVERTED INTEGER VALUE (SECONDS)
157E 83C9 A         LD      AC2,AEPCY       ; MAXIMUM CYCLE LENGTH TEST (SECONDS)
157F E600 A         SKG     AC1,D0(AC2)     ; SKIP IF .GT. MAXIMUM
1580 2107 A         JMP     .+8
1581        ;
1581        ;                   ... RECORD CYCLE LENGTH INPUT ERROR ...
1581 8CD2 A         LD      AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
1582 2C51 A         JSR     @AERE8          ; AC1 = ERROR DATUM
1583 4C01 A         LI      AC0,C1
1584 B0CE A         ST      AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
1585 94C9 A         LD      AC1,@AEPCY      ; SET CYCLE LENGTH TO MAXIMUM (SECONDS)
1586 8CC0 A         LD      AC3,AEPD2
1587 A700 A         ST      AC1,D0(AC3)     ; SAVE MAXIMUM CYCLE LENGTH (SECONDS)
1588        ;
1588 2C47 A         JSR     @AEM10          ; CONVERT TO 0.1 SECOND UNITS IN AC1
1589 8CC3 A         LD      AC3,AEPCD2
158A A703 A         ST      AC1,D3(AC3)     ; SAVE CYCLE LENGTH (0.1 SEC)
158B        ;
158B 88C5 A         LD      AC2,AEPFT       ; MINIMUM CYCLE LENGTH TEST (0.1 SEC)
158C 8201 A         LD      AC0,D1(AC2)     ; SUM OF FIXED INTERVALS
158D 5001 A         CAI     AC0,C1          ; NEGATE
158E 3400 A         RADD    AC1,AC0         ; AC0 = ADJ. TIME IF NON-NEG.
158F 1B02 A         BOC     CCZN,.+3        ; BRANCH IF CYCLE .LE. MINIMUM
1590 A301 A         ST      AC0,D1(AC3)     ; SAVE ADJUSTABLE TIME IN ADJ. INT. 2
1591 2109 A         JMP     .+10
1592        ;
1592        ;                   ... CYCLE LENGTH INPUT ERROR (TOO SHORT) ...
1592 4C00 A         LI      AC0,C0          ; NO ADJUSTABLE TIME
1593 A301 A         ST      AC0,D1(AC3)     ; SAVE 0 IN ADJUSTABLE INTERVAL 2
1594        ;
1594 8200 A         LD      AC0,D0(AC2)     ; RESTORE MIN. SUM OF FIXED INTERVALS
1595 8CC3 A         LD      AC3,AEPCD2
1596 A303 A         ST      AC0,D3(AC3)     ; STORE MIN CYCLE (0.1 SEC)
1597 8CD2 A         LD      AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
1598 2C51 A         JSR     @AERE8          ; AC1 = ERROR DATUM
1599 4C01 A         LI      AC0,C1
159A B0CE A         ST      AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
159B        ;
159B        ;                   ... COMPUTE MAX DWELL TIME FOR SYNCH (24% OF CYCLE) ...
159B 8CC0 A         LD      AC3,AEPD2
159C 8700 A         LD      AC1,D0(AC3)     ; CYCLE LENGTH (SECONDS)
```

```
159D 2C48 A        JSR    @AEM24          ; MULTIPLY BY 24
159E 4C00 A        LI     AC0,C0          ; PREPARE FOR DIVIDE
159F 88CC A        LD     AC2,AE10        ; *10/100 TO GET 0.1 SEC UNITS
15A0 0690 A        DIV    D0(AC2)
15A1 0000 A
15A2 8CC3 A        LD     AC3,AEPCD2
15A3 A708 A        ST     AC1,D8(AC3)     ; MAX DWELL 24% OF CYCLE (0.1 SEC)
15A4               ;
15A4               ;          ... COMPUTE START PULSE DURATION (3% OF CYCLE) ...
15A4 5DFD A        SHR    AC1,D3          ; DIVIDE BY 8
15A5 A707 A        ST     AC1,D7(AC3)     ; START PULSE DURATION 3% OF CYCLE
15A6 2105 A        JMP    .+6
15A7               ;
15A7               ;          ... RECORD CYCLE LENGTH VALIDITY ERROR ...
15A7               EPP20:
15A7 8CD2 A        LD     AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
15A8 2C4A A        JSR    @AERE1          ; AC1 = ERROR DATUM
15A9 4C01 A        LI     AC0,C1
15AA B0CE A        ST     AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
15AB B0D0 A        ST     AC0,@AEPDVE     ; SET ERROR FLAG FOR DATA VALIDITY
15AC               ;
15AC               ;          ... PROCESS OFFSET 1 ...
15AC 88BD A        LD     AC2,ABPD2
15AD 8201 A        LD     AC0,D1(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
15AE 2C46 A        JSR    @AEPBI3         ; CONVERT DATUM TO INTEGER IN AC1
15AF 2110 A        JMP    EPP21           ; DATA VALIDITY ERROR
15B0 8CC0 A        LD     AC3,AEPD2
15B1 A701 A        ST     AC1,D1(AC3)     ; SAVE CONVERTED VALUE (SECONDS)
15B2 E700 A        SKG    AC1,D0(AC3)     ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
15B3 2106 A        JMP    .+7
15B4               ;
15B4               ;          ... RECORD OFFSET INPUT ERROR ...
15B4 8CD2 A        LD     AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
15B5 2C52 A        JSR    @AERE9          ; AC1 = ERROR DATUM
15B6 4C01 A        LI     AC0,C1
15B7 B0CE A        ST     AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
15B8 4DF6 A        LI     AC1,CM10        ; SET OUT OF RANGE RESET TO -1 SECOND
15B9 2103 A        JMP    .+4
15BA               ;
15BA               ;          ... COMPUTE RESET ...
15BA D700 A        SUB    AC1,D0(AC3)     ; OFFSET - CYCLE
15BB 5101 A        CAI    AC1,C1          ; NEGATE
15BC 2C47 A        JSR    @AEM10          ; MULTIPLY AC1 BY 10
15BD 8CC3 A        LD     AC3,AEPCD2
15BE A704 A        ST     AC1,D4(AC3)     ; SAVE RESET (0.1 SECONDS)
15BF 2105 A        JMP    .+6
15C0               ;
15C0               ;          ... RECORD OFFSET DATA VALIDITY ERROR ...
15C0 8CD2 A EPP21: LD     AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
15C1 2C4B A        JSR    @AERE2          ; AC1 = ERROR DATUM
15C2 4C01 A        LI     AC0,C1
15C3 B0CE A        ST     AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
```

```
15C4 B0D0 A        ST    AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
15C5          ;
15C5          ;                  ... PROCESS OFFSET 2 ...
15C5 8BBD A        LD    AC2,ABPD2
15C6 8202 A        LD    AC0,D2(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
15C7 2C46 A        JSR   @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
15C8 2110 A        JMP   EPP22            ; DATA VALIDITY ERROR
15C9 8CC0 A        LD    AC3,AEPD2
15CA A702 A        ST    AC1,D2(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
15CB E700 A        SKG   AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
15CC 2106 A        JMP   .+7
15CD          ;
15CD          ;                  ... RECORD OFFSET INPUT ERROR ...
15CD 8CD2 A        LD    AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15CE 2C53 A        JSR   @AERE10          ; AC1 = ERROR DATUM
15CF 4C01 A        LI    AC0,C1
15D0 B0CE A        ST    AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15D1 4DF6 A        LI    AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
15D2 2103 A        JMP   .+4
15D3          ;
15D3          ;                  ... COMPUTE RESET ...
15D3 D700 A        SUB   AC1,D0(AC3)      ; OFFSET - CYCLE
15D4 5101 A        CAI   AC1,C1           ; NEGATE
15D5 2C47 A        JSR   @AEM10           ; MULTIPLY AC1 BY 10
15D6 8CC3 A        LD    AC3,AEPCD2
15D7 A705 A        ST    AC1,D5(AC3)      ; SAVE RESET (0.1 SECONDS)
15D8 2105 A        JMP   .+6
15D9          ;
15D9          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
15D9 8CD2 A EPP22: LD    AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15DA 2C4C A        JSR   @AERE3           ; AC1 = ERROR DATUM
15DB 4C01 A        LI    AC0,C1
15DC B0CE A        ST    AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15DD B0D0 A        ST    AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
15DE          ;
15DE          ;                  ... PROCESS OFFSET 3 ...
15DE 8BBD A        LD    AC2,ABPD2
15DF 8203 A        LD    AC0,D3(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
15E0 2C46 A        JSR   @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
15E1 2110 A        JMP   EPP23            ; DATA VALIDITY ERROR
15E2 8CC0 A        LD    AC3,AEPD2
15E3 A703 A        ST    AC1,D3(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
15E4 E700 A        SKG   AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
15E5 2106 A        JMP   .+7
15E6          ;
15E6          ;                  ... RECORD OFFSET INPUT ERROR ...
15E6 8CD2 A        LD    AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15E7 2C54 A        JSR   @AERE11          ; AC1 = ERROR DATUM
15E8 4C01 A        LI    AC0,C1
15E9 B0CE A        ST    AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15EA 4DF6 A        LI    AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
15EB 2103 A        JMP   .+4
```

```
2389 15FC          ;
2390 15EC          ;                    ... COMPUTE RESET ...
2391 15EC D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
2392 15ED 5101 A          CAI     AC1,C1           ; NEGATE
2393 15EE 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
2394 15EF 8CC3 A          LD      AC3,AEPCD2
2395 15F0 A706 A          ST      AC1,D6(AC3)      ; SAVE RESET (0.1 SECONDS)
2396 15F1 2105 A          JMP     .+6
2397 15F2          ;
2398 15F2          ;                    ... RECORD OFFSET DATA VALIDITY ERROR ...
2399 15F2 8CD2 A EPP23:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
2400 15F3 2C4D A          JSR     @AERE4           ; AC1 = ERROR DATUM
2401 15F4 4C01 A          LI      AC0,C1
2402 15F5 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
2403 15F6 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
2404 15F7          ;
2405 15F7          ;                    ... PROCESS MSG SPLIT % ...
2406 15F7 88BD A          LD      AC2,ABPD2
2407 15F8 8204 A          LD      AC0,D4(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2408 15F9 2C45 A          JSR     @AEPBI2          ; CONVERT DATUM TO INTEGER IN AC1
2409 15FA 2114 A          JMP     EPP24            ; DATA VALIDITY ERROR
2410 15FB 8CC0 A          LD      AC3,AEPD2
2411 15FC A704 A          ST      AC1,D4(AC3)      ; SAVE CONVERTED VALUE (%)
2412 15FD          ;
2413 15FD 88C3 A          LD      AC2,AEPCD2
2414 15FE 8201 A          LD      AC0,D1(AC2)      ; FETCH ADJUSTABLE TIME
2415 15FF 1B01 A          BOC     CCZN,.+2         ; TEST FOR POSITIVE
2416 1600 2102 A          JMP     .+3
2417 1601          ;
2418 1601 4D00 A          LI      AC1,C0           ; STORE 0 FOR BOTH ADJ. INTS.
2419 1602 2106 A          JMP     .+7
2420 1603          ;
2421 1603 4C00 A          LI      AC0,C0           ; PREPARE FOR MULTIPLY
2422 1604 0680 A          MPY     D1(AC2)          ; MSG(%) * ADJ. TIME (0.1 SEC)
     1605 0001 A
2423 1606 88CB A          LD      AC2,AE100        ; DIVIDE BY 100
2424 1607 0690 A          DIV     D0(AC2)
     1608 0000 A
2425 1609          ;
2426 1609 88C3 A          LD      AC2,AEPCD2
2427 160A A600 A          ST      AC1,D0(AC2)      ; STORE ADJ. INTERVAL 1
2428 160B 5101 A          CAI     AC1,C1           ; SUBTRACT ADJ. INT. 1 FROM TOTAL
2429 160C C601 A          ADD     AC1,D1(AC2)
2430 160D A601 A          ST      AC1,D1(AC2)      ; STORE ADJ. INT. 2
2431 160E 2105 A          JMP     .+6
2432 160F          ;
2433 160F          ;                    ... RECORD MSG DATA VALIDITY ERROR ...
2434 160F 8CD2 A EPP24:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
2435 1610 2C4E A          JSR     @AERE5           ; AC1 = ERROR DATUM
2436 1611 4C01 A          LI      AC0,C1
2437 1612 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
2438 1613 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
```

```
1614          ;
1614          ;                    ... PROCESS INT1 ...
1614 88BD A          LD      AC2,ABPD2
1615 8205 A          LD      AC0,D5(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1616 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
1617 210E A          JMP     EPP25            ; DATA VALIDITY ERROR
1618 8CC0 A          LD      AC3,AEPD2
1619 A705 A          ST      AC1,D5(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
161A 88CA A          LD      AC2,AEPIN1
161B E600 A          SKG     AC1,D0(AC2)      ; SKIP IF INT1 .GT. LIMIT
161C 2105 A          JMP     .+6
161D          ;
161D          ;                    ... RECORD INT1 INPUT ERROR ...
161D 8CD2 A          LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
161E 2C55 A          JSR     @AERE12          ; AC1 = ERROR DATUM
161F 4C01 A          LI      AC0,C1
1620 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
1621 94CA A          LD      AC1,@AEPIN1      ; SET MAXIMUM EXTENSION TO LIMIT (SEC)
1622          ;
1622 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
1623 8CC3 A          LD      AC3,AEPCD2
1624 A702 A          ST      AC1,D2(AC3)      ; STORE MAXIMUM EXTENSION (0.1 SEC)
1625 2105 A          JMP     .+6
1626          ;
1626          ;                    ... RECORD INT1 DATA VALIDITY ERROR ...
1626 8CD2 A EPP25:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1627 2C4F A          JSR     @AERE6           ; AC1 = ERROR DATUM
1628 4C01 A          LI      AC0,C1
1629 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
162A B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
162B          ;
162B 0200 A          RTS     D0               ; RETURN FROM EPP2
162C          ;
```

```
162C                .LIST   LPP
162C          ;
162C          ;                 ... PROCESS PROG. PANEL DATA FOR PLAN 3 ...
162C          EPP3:
162C          ;
162C          ;                 ... PROCESS CYCLE LENGTH ...
162C 885E A         LD      AC2,AEPD3
162D 8200 A         LD      AC0,D0(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
162E 2C46 A         JSR     @AEPB13          ; CONVERTED VALUE RETURNED IN AC1
162F 212B A         JMP     EPP30            ; DATA VALIDITY ERROR
1630 8CC1 A         LD      AC3,AEPD3
1631 A700 A         ST      AC1,D0(AC3)      ; SAVE CONVERTED INTEGER VALUE (SECONDS)
1632 83C9 A         LD      AC2,AEPCY        ; MAXIMUM CYCLE LENGTH TEST (SECONDS)
1633 E600 A         SKG     AC1,D0(AC2)      ; SKIP IF .GT. MAXIMUM
1634 2107 A         JMP     .+8
1635          ;
1635          ;                 ... RECORD CYCLE LENGTH INPUT ERROR ...
1635 8CD3 A         LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
1636 2C51 A         JSR     @AERE8           ; AC1 = ERROR DATUM
1637 4C01 A         LI      AC0,C1
1638 B0CF A         ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
1639 94C9 A         LD      AC1,@AEPCY       ; SET CYCLE LENGTH TO MAXIMUM (SECONDS)
163A 8CC1 A         LD      AC3,AEPD3
163B A700 A         ST      AC1,D0(AC3)      ; SAVE MAXIMUM CYCLE LENGTH (SECONDS)
163C          ;
163C 2C47 A         JSR     @AEM10           ; CONVERT TO 0.1 SECOND UNITS IN AC1
163D 8CC4 A         LD      AC3,AEPCD3
163E A703 A         ST      AC1,D3(AC3)      ; SAVE CYCLE LENGTH (0.1 SEC)
163F          ;
163F 88C5 A         LD      AC2,AEPFT        ; MINIMUM CYCLE LENGTH TEST (0.1 SEC)
1640 8202 A         LD      AC0,D2(AC2)      ; SUM OF FIXED INTERVALS
1641 5001 A         CAI     AC0,C1           ; NEGATE
1642 3400 A         RADD    AC1,AC0          ; AC0 = ADJ. TIME IF NON-NEG.
1643 1B02 A         BOC     CCZN,.+3         ; BRANCH IF CYCLE .LE. MINIMUM
1644 A301 A         ST      AC0,D1(AC3)      ; SAVE ADJUSTABLE TIME IN ADJ. INT. 2
1645 2109 A         JMP     .+10
1646          ;
1646          ;                 ... CYCLE LENGTH INPUT ERROR (TOO SHORT) ...
1646 4C00 A         LI      AC0,C0           ; NO ADJUSTABLE TIME
1647 A301 A         ST      AC0,D1(AC3)      ; SAVE 0 IN ADJUSTABLE INTERVAL 2
1648          ;
1648 8200 A         LD      AC0,D0(AC2)      ; RESTORE MIN. SUM OF FIXED INTERVALS
1649 8CC4 A         LD      AC3,AEPCD3
164A A303 A         ST      AC0,D3(AC3)      ; STORE MIN CYCLE (0.1 SEC)
164B 8CD3 A         LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
164C 2C51 A         JSR     @AERE8           ; AC1 = ERROR DATUM
164D 4C01 A         LI      AC0,C1
164E B0CF A         ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
164F          ;
164F          ;                 ... COMPUTE MAX DWELL TIME FOR SYNCH (24% OF CYCLE) ...
164F 8CC1 A         LD      AC3,AEPD3
1650 8700 A         LD      AC1,D0(AC3)      ; CYCLE LENGTH (SECONDS)
```

```
1651 2C48 A          JSR    @AEM24        ; MULTIPLY BY 24
1652 4C00 A          LI     AC0,C0        ; PREPARE FOR DIVIDE
1653 8BCC A          LD     AC2,AE10      ; *10/100 TO GET 0.1 SEC UNITS
1654 0690 A          DIV    D0(AC2)
1655 0000 A
1656 8CC4 A          LD     AC3,AEPCD3
1657 A708 A          ST     AC1,D8(AC3)   ; MAX DWELL 24% OF CYCLE (0.1 SEC)
1658          ;
1658          ;                ... COMPUTE START PULSE DURATION (3% OF CYCLE) ...
1658 5DFD A          SHR    AC1,D3        ; DIVIDE BY 8
1659 A707 A          ST     AC1,D7(AC3)   ; START PULSE DURATION 3% OF CYCLE
165A 2105 A          JMP    .+6
165B          ;
165B          ;                ... RECORD CYCLE LENGTH VALIDITY ERROR ...
165B          EPP30:
165B 8CD3 A          LD     AC3,AEPEF3    ; BASE ADDRESS OF ERROR BUFFER
165C 2C4A A          JSR    @AERE1        ; AC1 = ERROR DATUM
165D 4C01 A          LI     AC0,C1
165E B0CF A          ST     AC0,@AEP3E    ; SET ERROR FLAG FOR PLAN 3
165F B0D0 A          ST     AC0,@AEPDVE   ; SET ERROR FLAG FOR DATA VALIDITY
1660          ;
1660          ;                ... PROCESS OFFSET 1 ...
1660 8BBE A          LD     AC2,ABPD3
1661 8201 A          LD     AC0,D1(AC2)   ; RAW DATUM, BINARY-CODED DIGITS
1662 2C46 A          JSR    @AEPBI3       ; CONVERT DATUM TO INTEGER IN AC1
1663 2110 A          JMP    EPP31         ; DATA VALIDITY ERROR
1664 8CC1 A          LD     AC3,AEPD3
1665 A701 A          ST     AC1,D1(AC3)   ; SAVE CONVERTED VALUE (SECONDS)
1666 E700 A          SKG    AC1,D0(AC3)   ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
1667 2106 A          JMP    .+7
1668          ;
1668          ;                ... RECORD OFFSET INPUT ERROR ...
1668 8CD3 A          LD     AC3,AEPEF3    ; BASE ADDRESS OF ERROR BUFFER
1669 2C52 A          JSR    @AERE9        ; AC1 = ERROR DATUM
166A 4C01 A          LI     AC0,C1
166B B0CF A          ST     AC0,@AEP3E    ; SET ERROR FLAG FOR PLAN 3
166C 4DF6 A          LI     AC1,CM10      ; SET OUT OF RANGE RESET TO -1 SECOND
166D 2103 A          JMP    .+4
166E          ;
166E          ;                ... COMPUTE RESET ...
166E D700 A          SUB    AC1,D0(AC3)   ; OFFSET - CYCLE
166F 5101 A          CAI    AC1,C1        ; NEGATE
1670 2C47 A          JSR    @AEM10        ; MULTIPLY AC1 BY 10
1671 8CC4 A          LD     AC3,AEPCD3
1672 A704 A          ST     AC1,D4(AC3)   ; SAVE RESET (0.1 SECONDS)
1673 2105 A          JMP    .+6
1674          ;
1674          ;                ... RECORD OFFSET DATA VALIDITY ERROR ...
1674 8CD3 A EPP31:   LD     AC3,AEPEF3    ; BASE ADDRESS OF ERROR BUFFER
1675 2C4B A          JSR    @AERE2        ; AC1 = ERROR DATUM
1676 4C01 A          LI     AC0,C1
1677 B0CF A          ST     AC0,@AEP3E    ; SET ERROR FLAG FOR PLAN 3
```

```
1678 B0D0 A        ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1679          ;
1679          ;                  ... PROCESS OFFSET 2 ...
1679 88BE A        LD      AC2,ABPD3
1679A 8202 A       LD      AC0,D2(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1679B 2C46 A       JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
1679C 2110 A       JMP     EPP32            ; DATA VALIDITY ERROR
1679D 8CC1 A       LD      AC3,AEPD3
1679E A702 A       ST      AC1,D2(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
1679F E700 A       SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
1680 2106 A        JMP     .+7
1681          ;
1681          ;                  ... RECORD OFFSET INPUT ERROR ...
1681 8CD3 A        LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
1682 2C53 A        JSR     @AERE10          ; AC1 = ERROR DATUM
1683 4C01 A        LI      AC0,C1
1684 B0CF A        ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
1685 4DF6 A        LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
1686 2103 A        JMP     .+4
1687          ;
1687          ;                  ... COMPUTE RESET ...
1687 D700 A        SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
1688 5101 A        CAI     AC1,C1           ; NEGATE
1689 2C47 A        JSR     @AEM10           ; MULTIPLY AC1 BY 10
168A 8CC4 A        LD      AC3,AEPCD3
168B A705 A        ST      AC1,D5(AC3)      ; SAVE RESET (0.1 SECONDS)
168C 2105 A        JMP     .+6
168D          ;
168D          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
168D 8CD3 A EPP32: LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
168E 2C4C A        JSR     @AERE3           ; AC1 = ERROR DATUM
168F 4C01 A        LI      AC0,C1
1690 B0CF A        ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
1691 B0D0 A        ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1692          ;
1692          ;                  ... PROCESS OFFSET 3 ...
1692 88BE A        LD      AC2,ABPD3
1693 8203 A        LD      AC0,D3(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1694 2C46 A        JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
1695 2110 A        JMP     EPP33            ; DATA VALIDITY ERROR
1696 8CC1 A        LD      AC3,AEPD3
1697 A703 A        ST      AC1,D3(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
1698 E700 A        SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
1699 2106 A        JMP     .+7
169A          ;
169A          ;                  ... RECORD OFFSET INPUT ERROR ...
169A 8CD3 A        LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
169B 2C54 A        JSR     @AERE11          ; AC1 = ERROR DATUM
169C 4C01 A        LI      AC0,C1
169D B0CF A        ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
169E 4DF6 A        LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
169F 2103 A        JMP     .+4
```

```
16A0          ;
16A0          ;                    ... COMPUTE RESET ...
16A0 D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
16A1 5101 A          CAI     AC1,C1           ; NEGATE
16A2 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
16A3 8CC4 A          LD      AC3,AEPCD3
16A4 A706 A          ST      AC1,D6(AC3)      ; SAVE RESET (0.1 SECONDS)
16A5 2105 A          JMP     .+6
16A6          ;
16A6          ;                    ... RECORD OFFSET DATA VALIDITY ERROR ...
16A6 8CD3 A EPP33:   LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
16A7 2C4D A          JSR     @AERE4           ; AC1 = ERROR DATUM
16A8 4C01 A          LI      AC0,C1
16A9 B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
16AA B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
16AB          ;
16AB          ;                    ... PROCESS MSG SPLIT % ...
16AB 88BE A          LD      AC2,ABPD3
16AC 8204 A          LD      AC0,D4(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
16AD 2C45 A          JSR     @AEPBI2          ; CONVERT DATUM TO INTEGER IN AC1
16AE 2114 A          JMP     EPP34            ; DATA VALIDITY ERROR
16AF 8CC1 A          LD      AC3,AEPD3
16B0 A704 A          ST      AC1,D4(AC3)      ; SAVE CONVERTED VALUE (%)
16B1          ;
16B1 88C4 A          LD      AC2,AEPCD3
16B2 8201 A          LD      AC0,D1(AC2)      ; FETCH ADJUSTABLE TIME
16B3 1B01 A          BOC     CCZN,.+2         ; TEST FOR POSITIVE
16B4 2102 A          JMP     .+3
16B5          ;
16B5 4D00 A          LI      AC1,C0           ; STORE 0 FOR BOTH ADJ. INTS.
16B6 2106 A          JMP     .+7
16B7          ;
16B7 4C00 A          LI      AC0,C0           ; PREPARE FOR MULTIPLY
16B8 0680 A          MPY     D1(AC2)          ; MSG(%) * ADJ. TIME (0.1 SEC)
16B9 0001 A
16BA 88CB A          LD      AC2,AE100        ; DIVIDE BY 100
16BB 0690 A          DIV     D0(AC2)
16BC 0000 A
16BD          ;
16BD 88C4 A          LD      AC2,AEPCD3
16BE A600 A          ST      AC1,D0(AC2)      ; STORE ADJ. INTERVAL 1
16BF 5101 A          CAI     AC1,C1           ; SUBTRACT ADJ. INT. 1 FROM TOTAL
16C0 C601 A          ADD     AC1,D1(AC2)
16C1 A601 A          ST      AC1,D1(AC2)      ; STORE ADJ. INT. 2
16C2 2105 A          JMP     .+6
16C3          ;
16C3          ;                    ... RECORD MSG DATA VALIDITY ERROR ...
16C3 8CD3 A EPP34:   LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
16C4 2C4E A          JSR     @AERE5           ; AC1 = ERROR DATUM
16C5 4C01 A          LI      AC0,C1
16C6 B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
16C7 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
```

```
16C8          ;
16C8          ;                    ... PROCESS INT1 ...
16C8 88BE A          LD      AC2,ABPD3
16C9 8205 A          LD      AC0,D5(AC2)       ; RAW DATUM, BINARY-CODED DIGITS
16CA 2C46 A          JSR     @AEPBI3           ; CONVERT DATUM TO INTEGER IN AC1
16CB 210E A          JMP     EPP35             ; DATA VALIDITY ERROR
16CC 8CC1 A          LD      AC3,AEPD3
16CD A705 A          ST      AC1,D5(AC3)       ; SAVE CONVERTED VALUE (SECONDS)
16CE 88CA A          LD      AC2,AEPIN1
16CF E600 A          SKG     AC1,D0(AC2)       ; SKIP IF INT1 .GT. LIMIT
16D0 2105 A          JMP     .+6
16D1          ;
16D1          ;                    ... RECORD INT1 INPUT ERROR ...
16D1 8CD3 A          LD      AC3,AEPEF3        ; BASE ADDRESS OF ERROR BUFFER
16D2 2C55 A          JSR     @AERE12           ; AC1 = ERROR DATUM
16D3 4C01 A          LI      AC0,C1
16D4 B0CF A          ST      AC0,@AEP3E        ; SET ERROR FLAG FOR PLAN 3
16D5 94CA A          LD      AC1,@AEPIN1       ; SET MAXIMUM EXTENSION TO LIMIT (SEC)
16D6          ;
16D6 2C47 A          JSR     @AEM10            ; MULTIPLY AC1 BY 10
16D7 8CC4 A          LD      AC3,AEPCD3
16D8 A702 A          ST      AC1,D2(AC3)       ; STORE MAXIMUM EXTENSION (0.1 SEC)
16D9 2105 A          JMP     .+6
16DA          ;
16DA          ;                    ... RECORD INT1 DATA VALIDITY ERROR ...
16DA 8CD3 A EPP35:   LD      AC3,AEPEF3        ; BASE ADDRESS OF ERROR BUFFER
16DB 2C4F A          JSR     @AERE6            ; AC1 = ERROR DATUM
16DC 4C01 A          LI      AC0,C1
16DD B0CF A          ST      AC0,@AEP3E        ; SET ERROR FLAG FOR PLAN 3
16DE B0D0 A          ST      AC0,@AEPDVE       ; SET ERROR FLAG FOR DATA VALIDITY
16DF          ;
16DF 0200 A          RTS     D0                ; RETURN FROM EPP3
16E0          ;
```

```
2714 16E0                .LIST   LPP
2715 16F0          ;
2716 16E0          ;                    ... UPDATE PROG. PANEL DATA ...
2717 16E0          EPUP:
2718 16E0          ;                    ... UPDATE ADJUSTABLE INTERVALS ...
2719 16E0          ;
2720 16E0          ;                                  DIAL 1
2721 16E0 9893 A          LD      AC2,@ATAI1              ; FIRST ADJUSTABLE INTERVAL
2722 16E1 C88B A          ADD     AC2,ATPM1
2723 16E2 8CC2 A          LD      AC3,AEPCD1
2724 16E3 8700 A          LD      AC1,D0(AC3)
2725 16E4 A600 A          ST      AC1,(AC2)
2726 16E5          ;
2727 16E5 9894 A          LD      AC2,@ATAI2              ; SECOND ADJUSTABLE INTERVAL
2728 16E6 C88B A          ADD     AC2,ATPM1
2729 16E7 8701 A          LD      AC1,D1(AC3)
2730 16E8 A600 A          ST      AC1,(AC2)
2731 16E9          ;                                  DIAL 2
2732 16E9 9893 A          LD      AC2,@ATAI1              ; FIRST ADJUSTABLE INTERVAL
2733 16EA C88C A          ADD     AC2,ATPM2
2734 16EB 8CC3 A          LD      AC3,AEPCD2
2735 16EC 8700 A          LD      AC1,D0(AC3)
2736 16ED A600 A          ST      AC1,(AC2)
2737 16EE          ;
2738 16EE 9894 A          LD      AC2,@ATAI2              ; SECOND ADJUSTABLE INTERVAL
2739 16EF C88C A          ADD     AC2,ATPM2
2740 16F0 8701 A          LD      AC1,D1(AC3)
2741 16F1 A600 A          ST      AC1,(AC2)
2742 16F2          ;                                  DIAL 3
2743 16F2 9893 A          LD      AC2,@ATAI1              ; FIRST ADJUSTABLE INTERVAL
2744 16F3 C88D A          ADD     AC2,ATPM3
2745 16F4 8CC4 A          LD      AC3,AEPCD3
2746 16F5 8700 A          LD      AC1,D0(AC3)
2747 16F6 A600 A          ST      AC1,(AC2)
2748 16F7          ;
2749 16F7 9894 A          LD      AC2,@ATAI2              ; SECOND ADJUSTABLE INTERVAL
2750 16F8 C88D A          ADD     AC2,ATPM3
2751 16F9 8701 A          LD      AC1,D1(AC3)
2752 16FA A600 A          ST      AC1,(AC2)
2753 16FB          ;
2754 16FB          ;                    ... UPDATE MSG SYNCH DWELL PARAMETERS ...
2755 16FB 8C89 A          LD      AC3,ATDWL       ; ADDRESS OF MAX DWELL TIMES
2756 16FC 88C2 A          LD      AC2,AEPCD1      ; ADDRESS OF CONVERTED DATA FOR DIAL 1
2757 16FD 8608 A          LD      AC1,D8(AC2)
2758 16FE A700 A          ST      AC1,D0(AC3)
2759 16FF          ;
2760 16FF 88C3 A          LD      AC2,AEPCD2      ; ADDRESS OF CONVERTED DATA FOR DIAL 2
2761 1700 8608 A          LD      AC1,D8(AC2)
2762 1701 A701 A          ST      AC1,D1(AC3)
2763 1702          ;
2764 1702 88C4 A          LD      AC2,AEPCD3      ; ADDRESS OF CONVERTED DATA FOR DIAL 3
2765 1703 8608 A          LD      AC1,D8(AC2)
```

```
1704 A702 A        ST      AC1,D2(AC3)
1705          ;
1705          ;                ... UPDATE MAX EXTENSION PREEMPT PARAMETERS ...
1705 88C2 A        LD      AC2,AEPCD1          ; DIAL 1
1706 8602 A        LD      AC1,D2(AC2)
1707 8C76 A        LD      AC3,APEMAX
1708 A701 A        ST      AC1,D1(AC3)
1709          ;
1709 88C3 A        LD      AC2,AEPCD2          ; DIAL 2
170A 8602 A        LD      AC1,D2(AC2)
170B A702 A        ST      AC1,D2(AC3)
170C          ;
170C 88C4 A        LD      AC2,AEPCD3          ; DIAL 3
170D 8602 A        LD      AC1,D2(AC2)
170E A703 A        ST      AC1,D3(AC3)
170F          ;
170F          ;                ... UPDATE SYNCHRONIZATION PARAMETERS ...
170F 88A1 A        LD      AC2,AESYD
1710 8CC2 A        LD      AC3,AEPCD1          ; DIAL 1
1711 2908 A        JSR     EPUPS
1712          ;
1712 4006 A        LI      AC1,C6          ; BUFFER LENGTH = 6
1713 3600 A        RADD    AC1,AC2         ; INCREMENT ADDRESS BY BUFFER LENGTH
1714 8CC3 A        LD      AC3,AEPCD2          ; DIAL 2
1715 2904 A        JSR     EPUPS
1716          ;
1716 3600 A        RADD    AC1,AC2         ; INCREMENT ADDRESS BY BUFFER LENGTH
1717 8CC4 A        LD      AC3,AEPCD3          ; DIAL 3
1718 2901 A        JSR     EPUPS
1719          ;
1719 0200 A        RTS     DO              ; RETURN FROM EPUP
171A          ;
171A          ;                                  COPY SYNC DATA ROUTINE
171A 8303 A EPUPS: LD      AC0,D3(AC3)
171B A200 A        ST      AC0,D0(AC2)     ; CYCLE LENGTH (0.1 SEC UNITS)
171C A205 A        ST      AC0,D5(AC2)     ; SET SYNCH DIAL TIMER TO CYCLE LENGTH
171D          ;
171D 8304 A        LD      AC0,D4(AC3)
171E A201 A        ST      AC0,D1(AC2)     ; RESET 1 (0.1 SEC UNITS)
171F          ;
171F 8305 A        LD      AC0,D5(AC3)
1720 A202 A        ST      AC0,D2(AC2)     ; RESET 2 (0.1 SEC UNITS)
1721          ;
1721 8306 A        LD      AC0,D6(AC3)
1722 A203 A        ST      AC0,D3(AC2)     ; RESET 3 (0.1 SEC UNITS)
1723          ;
1723 8307 A        LD      AC0,D7(AC3)
1724 A204 A        ST      AC0,D4(AC2)     ; START PULSE DURATION (0.1 SEC UNITS)
1725          ;
1725 0200 A        RTS     DO              ; RETURN FROM EPUPS
1726          ;
```

```
1726          .LIST   LPP
1726     ;                ... CONVERT BINARY CODED DATA TO INTEGER ...
1726     ;            AT ENTRY, AC0 CONTAINS BINARY CODED DIGITS
1726     ;            ENTRY POINT:
1726     ;                        EPBI3 - AC0 HAS 0,N2,N1,N0
1726     ;                        EPBI2 - AC0 HAS 0, 0,N1,N0
1726     ;                        EPBI1 - AC0 HAS 0, 0, 0,N0
1726     ;            INTEGER VALUE RETURNED IN AC1
1726     EPBI:
1726     ;
1726 4E00 A EPBI3:  LI      AC2,C0
1727 3181 A         RCPY    AC0,AC1          ; PROCESS THIRD DIGIT (N2)
1728 651E A         AND     AC1,H0F00        ; EXTRACT THIRD DIGIT N2*256
1729 E520 A         SKG     AC1,H0900     ; TEST FOR VALID DIGIT
172A 2101 A         JMP     .+2
172B 211A A         JMP     EPBIR+1       ; ERROR EXIT
172C 5DFE A         SHR     AC1,D2
172D 3600 A         RADD    AC1,AC2          ; AC2 = AC2 + N2*64
172E 5DFF A         SHR     AC1,D1
172F 3600 A         RADD    AC1,AC2          ; AC2 = AC2 + N2*(64 + 32)
1730 5DFD A         SHR     AC1,D3
1731 3600 A         RADD    AC1,AC2          ; AC2 = AC2 + N2*100
1732 2101 A         JMP     .+2
1733 4E00 A EPBI2:  LI      AC2,C0
1734 3181 A         RCPY    AC0,AC1          ; PROCESS SECOND DIGIT (N1)
1735 6512 A         AND     AC1,H00F0        ; EXTRACT SECOND DIGIT N1*16
1736 E514 A         SKG     AC1,H0090     ; TEST FOR VALID DIGIT
1737 2101 A         JMP     .+2
1738 210D A         JMP     EPBIR+1       ; ERROR EXIT
1739 5DFF A         SHR     AC1,D1
173A 3600 A         RADD    AC1,AC2          ; AC2 = AC2 + N1*8
173B 5DFE A         SHR     AC1,D2
173C 3600 A         RADD    AC1,AC2          ; AC2 = AC2 + N1*10
173D 2101 A         JMP     .+2
173E 4E00 A EPBI1:  LI      AC2,C0
173F 3181 A         RCPY    AC0,AC1          ; PROCESS FIRST DIGIT (N0)
1740 6508 A         AND     AC1,H000F        ; EXTRACT FIRST DIGIT N0*1
1741 E50A A         SKG     AC1,H0009     ; TEST FOR VALID DIGIT
1742 2101 A         JMP     .+2
1743 2102 A         JMP     EPBIR+1       ; ERROR EXIT
1744 3900 A         RADD    AC2,AC1          ; AC1 = AC2 + N0, FINAL INTEGER
1745     ;
1745 0201 A EPBIR:  RTS     D1            ; RETURN FROM EPBI
1746 0200 A         RTS     D0            ; ERROR RETURN FROM EPBI
1747     ;
1747     ;                ... LOCAL DATA ...
1747 0F00 A H0F00:  .WORD   X'0F00
1748 00F0 A H00F0:  .WORD   X'00F0
1749 000F A H000F:  .WORD   X'000F
174A 0900 A H0900:  .WORD   X'0900        ; MAX VALUE FOR N2*256
174B 0090 A H0090:  .WORD   X'0090        ; MAX VALUE FOR N1*16
174C 0009 A H0009:  .WORD   X'0009        ; MAX VALUE FOR N0*1
```

2871 174D ;

```
174D                    .LIST LPP
174D            ;
174D            ;
174D            ;                    ... ROUTINES TO MULTIPLY AN INTEGER BY A CONSTANT ...
174D            ;
174D            ;                    UPON ENTRY, AC1 CONTAINS THE INTEGER
174D            ;                                AC0 USED FOR TEMPORARY STORAGE
174D            ;
174D            ;                    UPON RETURN, AC1 CONTAINS RESULT
174D            ;
174D            ;                    ... MULTIPLY BY 10 ...
174D            EM10:
174D 5D01 A             SHL     AC1,D1          ; INTEGER*2
174E 3481 A             RCPY    AC1,AC0         ; SAVE
174F 5D02 A             SHL     AC1,D2          ; INTEGER*8
1750 3100 A             RADD    AC0,AC1         ; INTEGER*10
1751 0200 A             RTS     D0              ; RETURN FROM EM10
1752            ;
1752            ;                    ... MULTIPLY BY 24 ...
1752            EM24:
1752 5D03 A             SHL     AC1,D3          ; INTEGER*8
1753 3481 A             RCPY    AC1,AC0         ; SAVE
1754 5D01 A             SHL     AC1,D1          ; INTEGER*16
1755 3100 A             RADD    AC0,AC1         ; INTEGER*24
1756 0200 A             RTS     D0              ; RETURN FROM EM24
```

```
2900 1757                .PAGE
2901 1757                .ASECT
2902 1757 1800 A         .=X'1800
2903 1800              ;
```

```
1800          .LIST  LTP
1800     ;*****************************************************************************
1800     ;
1800     TPEX:
1800 2100 A        JMP     .+1                  ; JSR @ACOTF CAN BE INSERTED HERE
1801 9484 A        LD      AC1,@ATIN            ; AC1=CURRENT INTERVAL INDEX
1802 8892 A        LD      AC2,ATHI             ;  AC2=LOC OF THI
1803 F600 A        SKNE    AC1,(AC2)            ;SKIP IF NOT TRANSFER/HOLD INTERVAL
1804 2101 A        JMP     .+2
1805 2108 A        JMP     TMSG
1806 9069 A        LD      AC0,@ABDIN           ; AC0 = BDIN
1807 075E A        SKBIT   D14                  ; SKIP IF TRANSFER HOLD
1808 2105 A        JMP     TMSG
1809 8C97 A        LD      AC3,ATCIW            ; AC3=LOC OF COMMAND INDEX WORD TABLE
180A CE00 A        ADD     AC3,(AC2)            ; AC3=LOC OF CIW FOR T/H INTERVAL
180B 8300 A        LD      AC0,(AC3)            ; AC0=CIW FOR T/H INTERVAL
180C B082 A        ST      AC0,@ATCIWC          ; TCIWC=AC0
180D 0200 A        RTS     D0              ; RETURN FROM TPEX (WAIT MODE)
180E F51E A TMSG:  SKNE    AC1,T0               ; SKIP IF NOT MSG
180F 2908 A        JSR     TSST                 ; JUMP TO TP SYNC START ROUTINE
1810 2101 A        JMP     .+2                  ; JUMP OVER RTS FOR TIMING MODE
1811 0200 A        RTS     D0              ; RETURN FROM TPEX (SYNCH DWELL MODE)
1812 9C85 A        LD      AC3,@ATINT           ;  AC2=LOC OF INTERVAL TIMER
1813 DD1A A        SUB     AC3,T1               ;  DECREMENT INTERVAL TIMER
1814 BC85 A        ST      AC3,@ATINT
1815 ED17 A        SKG     AC3,T0               ;  SKIP IF INTERVAL NOT DONE
1816 291A A        JSR     TNXT                 ; INITIATE NEXT TP INTERVAL
1817 0200 A        RTS     D0              ; RETURN FROM TPEX (TIMING MODE)
1818     ;
```

```
1818          .LIST   LTP
1818        ;*****************************************************************************
1818        ;
1818      TSST:
1818 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1819 9C86 A         LD      AC3,@ATFST              ;  AC3=VALUE OF WAIT FOR SYNC START FLAG
181A ED12 A         SKG     AC3,T0                  ; SKIP IF WAIT FOR SYNC START FLAG IS SET
181B 210D A         JMP     TRSET
181C 8889 A         LD      AC2,ATDWL               ; BASE ADDRESS FOR DWELL TIME MAXIMUMS
181D 9483 A         LD      AC1,@ATCDS              ; CURRENT DIAL SELECTION
181E 49FF A         AISZ    AC1,CM1                 ; COMPUTE ADDRESS FOR CURRENT DIAL
181F 3600 A         RADD    AC1,AC2                 ; SKIP IF DIAL 1
1820 8E00 A         LD      AC3,(AC2)               ; MAX DWELL TIME
1821        ;
1821        ;
1821 889C A         LD      AC2,AETST               ; AC2=LOC OF DWELL TIMER
1822 EE00 A         SKG     AC3,(AC2)               ; SKIP IF DWELL TIMER < DWELL MAX
1823 2105 A         JMP     TRSET                   ; JUMP TO RESET WAIT FOR SYNC.START FLAG
1824 4D01 A         LI      AC1,C1
1825 889D A         LD      AC2,AEPST               ; AC2=LOC OF START PULSE INDICATOR
1826 4C00 A         LI      AC0,C0
1827 F200 A         SKNE    AC0,(AC2)               ; SKIP IF START PULSE PRESENT
1828 0201 A         RTS     D1                  ; RETURN FROM TSST (SYNCH DWELL MODE)
1829 9086 A TRSET:  LD      AC0,@ATFST
182A 072F A         SETBIT  D15                     ; SET MSG RELEASE BIT
182B 6086 A         ST      AC0,@ATFST
182C 0200 A         RTS     D0                  ; RETURN FROM TSST (TIMING MODE)
182D                                            ;  LOCAL CONSTANTS
182D 0000 A T0:     .WORD   0
182E 0001 A T1:     .WORD   1
182F 0002 A T2:     .WORD   2
1830 0003 A T3:     .WORD   3
1831        ;
```

```
1831                .LIST   LTP
1831        ;*****************************************************************************
1831        ;
1831        TNXT:
1831 2100 A         JMP     .+1                 ; JSR @ACOTF CAN BE INSERTED HERE
1832 4D01 A         LI      AC1,C1
1833 8884 A         LD      AC2,ATIN            ; AC2= LOC OF CURRENT INTERVAL INDEX
1834 C600 A         ADD     AC1,(AC2)
1835 B484 A         ST      AC1,@ATIN           ; INCREMENT INTERVAL INDEX(TIN)
1836 4C00 A         LI      AC0,C0
1837 E200 A         SKG     AC0,(AC2)           ; SKIP IF TIN<0
1838 2101 A         JMP     .+2
1839 211F A         JMP     TOL                 ; TIN < 0
183A 8C95 A         LD      AC3,ATIMX           ; AC3=LOC OF MAX VALUE FOR TIN
183B E700 A         SKG     AC1,(AC3)           ; SKIP IF TIN>MAX
183C 2101 A         JMP     .+2
183D 211B A         JMP     TOL                 ; TIN > TIMX
183E 8C8F A         LD      AC3,ATCTP           ; AC3=LOC OF CURRENT TIMING PLAN
183F CE00 A         ADD     AC3,(AC2)           ; AC3=LOC OF CURRENT INTERVAL IN TP
1840 8300 A         LD      AC0,(AC3)           ; AC0=CURRENT INTERVAL LENGTH
1841 1102 A         BOC     CCEZ,TEZ            ; BRANCH IF INTERVAL IS 0
1842 180A A         BOC     CCZN,TLZ            ; BRANCH IF INTERVAL < 0
1843 210E A         JMP     TGZ                 ; INTERVAL >0
1844 8C92 A TEZ:    LD      AC3,ATHI            ; AC3=LOC OF TRANSFER HOLD INDEX
1845 F700 A         SKNE    AC1,(AC3)           ; SKIP IF NOT T H INTERVAL
1846 2101 A         JMP     .+2
1847 21E9 A         JMP     TNXT            ; CURRENT INTERVAL  IS 0 AND NOT TRANSFER
1848                                        ; HOLD--JUMP TO NEXT INTERVAL
1848 9069 A         LD      AC0,@ABDIN          ; AC0=ABDIN
1849 075E A         SKBIT   D14                 ; SKIP IF TRANSFER HOLD
184A 21E6 A         JMP     TNXT                ; NO TRANSFER HOLD, SKIP TO NEXT INTERVAL
184B 8300 A         LD      AC0,(AC3)           ; AC0=CURRENT INTERVAL LENGTH
184C 2105 A         JMP     TGZ                 ; TRANSFER HOLD, TREAT AS FINITE INTERVAL
184D                                        ; TINT<0 SIGNALS END OF CYCLE
184D 4D00 A TLZ:    LI      AC1,C0
184E B484 A         ST      AC1,@ATIN           ; SET INTERVAL INDEX TO 0
184F 2C2C A         JSR     @AEEOC              ; JUMP TO END OF CYCLE ROUTINE
1850 8C8F A         LD      AC3,ATCTP           ; AC3=LOC OF CURRENT TP
1851 8300 A         LD      AC0,(AC3)           ; AC0=LENGTH OF FIRST INTERVAL
1852 B085 A TGZ:    ST      AC0,@ATINT          ; INITIALIZE CURRENT INTERVAL TIMER
1853 8897 A         LD      AC2,ATCIW           ; AC2=LOC OF COMMAND INDEX WORD TABLE
1854 9C84 A         LD      AC3,@ATIN           ; AC3=VALUE OF CURRENT INT INDEX
1855 3E00 A         RADD    AC3,AC2             ; AC2=LOC OF CURRENT COMMAND INDEX WORD
1856 8200 A         LD      AC0,(AC2)           ; AC0=VALUE OF COMMAND INDEX WORD
1857 B082 A         ST      AC0,@ATCIWC         ;TCIWC=VALUE OF COMMAND INDEX WORD
1858 0200 A         RTS     D0                  ; RETURN FROM TNXT
1859        TOL:                                ; INTERVAL INDEX OUT OF LIMITS, RECORD ERROR
1859                                            ;     AND DISCONNECT
1859 8890 A         LD      AC2,ATERR
185A 9084 A         LD      AC0,@ATIN
185B A204 A         ST      AC0,D4(AC2)         ; STORE ERROR WORD IN ERROR BUFFER
185C 8203 A         LD      AC0,D3(AC2)
```

TIMING PLAN CPC - INITIATE NEXT INTERVAL

```
3024 185D 4801 A        AISZ    AC0,C1              ; FETCH ERROR COUNTER AND INCREMENT
3025 185E 2100 A        JMP     .+1
3026 185F A203 A        ST      AC0,D3(AC2)
3027 1860 8200 A        LD      AC0,D0(AC2)
3028 1861 0721 A        SETBIT  D1                  ; SET ERROR FLAG BIT 1
3029 1862 A200 A        ST      AC0,D0(AC2)
3030 1863       ;
3031 1863 2C2F A        JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
3032 1864       ;
3033 1864 2100 A        JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
3034 1865 21FE A        JMP     .-1
3035 1866       ;
```

```
1866                .LIST  LTP
1866          ;*****************************************************************************
1866          ;
1866          TNCY:
1866          ;                                        ZERO PREQ AND PTER ENTRY TIMES
1866 4C00 A          LI      AC0,C0
1867 88BA A          LD      AC2,ACTST
1868 A200 A          ST      AC0,D0(AC2)
1869 A201 A          ST      AC0,D1(AC2)
186A          ;
186A          ;                                        TEST PREEMPT IN PROGRESS FLAG
186A 906E A          LD      AC0,@APIP                ; ACCESS PREEMPT IN PROGRESS FLAG
186B 1554 A          BOC     CCNZ,TNDC                ; BRANCH IF PREEMPT IN PROGRESS
186C                                                  ;    NO DIAL CHANGE PERMITTED
186C          ;
186C          ;                    ... MANUAL OVERRIDE TEST ...
186C 9CBB A          LD      AC0,@ABPPD          ; PROG PANEL SWITCHES
186D 075B A          SKBIT   D11                 ; SKIP IF MANUAL OVERRIDE IS ON
186E 2101 A          JMP     .+2                 ; MANUAL OVERRIDE OFF
186F 2116 A          JMP     TSYNCR
1870          ;
1870          ;                    ... PROG PANEL SYNCHRONIZATION SWITCHES ...
1870 8883 A TSYNC:   LD      AC2,ATCDS           ; ADDRESS OF CURRENT DIAL SELECT DATUM
1871          ;
1871 4D01 A          LI      AC1,C1
1872 F600 A          SKNE    AC1,(AC2)           ; SKIP IF NOT DIAL 1
1873 2101 A          JMP     .+2
1874 2103 A          JMP     TSD2                ; JUMP TO TEST DIAL 2
1875 0751 A          SKBIT   D1                  ; SKIP IF SYNCH SWITCH IS ON
1876 210F A          JMP     TSYNCR              ; JUMP TO RESET FLAG
1877 2110 A          JMP     TSYNCE
1878          ;
1878 4D02 A TSD2:    LI      AC1,C2
1879 F600 A          SKNE    AC1,(AC2)           ; SKIP IF NOT DIAL 2
187A 2101 A          JMP     .+2
187B 2103 A          JMP     TSD3                ; JUMP TO TEST DIAL 3
187C 0755 A          SKBIT   D5                  ; SKIP IF SYNCH SWITCH IS ON
187D 2108 A          JMP     TSYNCR              ; JUMP TO RESET FLAG
187E 2109 A          JMP     TSYNCE
187F          ;
187F 4D03 A TSD3:    LI      AC1,C3
1880 F600 A          SKNE    AC1,(AC2)           ; SKIP IF NOT DIAL 3
1881 2101 A          JMP     .+2
1882 2105 A          JMP     TSYNCE
1883 0759 A          SKBIT   D9                  ; SKIP IF SYNCH SWITCH IS ON
1884 2101 A          JMP     TSYNCR              ; JUMP TO RESET FLAG
1885 2102 A          JMP     TSYNCE
1886          ;
1886 4C00 A TSYNCR:  LI      AC0,C0
1887 BC86 A          ST      AC0,@ATFST          ; RESET 'WAIT FOR SYNCH START' FLAG
1888          ;
1888          TSYNCE:
```

```
3090 1888 906E A        LD      AC0,@APIP       ; FETCH PREEMPT IN PROGRESS FLAG
3091 1889 150C A        BOC     CCNZ,TPUE       ; BRANCH IF FLAG SET
3092 188A 90C6 A        LD      AC0,@AEPUF      ; FETCH PROG PANEL UPDATE FLAG
3093 188B 1501 A        BOC     CCNZ,.+2        ; BRANCH IF UPDATE FLAG IS SET
3094 188C 2109 A        JMP     TPUE
3095 188D         ;
3096 188D 2C44 A        JSR     @AEPUP          ; UPDATE PROG. PANEL DATA
3097 188E         ;
3098 188E 8CE9 A        LD      AC3,APPDU
3099 188F 2C57 A        JSR     @AXLOF1         ; ASSURE PENDING INPUT LIGHT IS OFF
3100 1890 4C00 A        LI      AC0,C0
3101 1891 B0C6 A        ST      AC0,@AEPUF      ; RESET PROG PANEL UPDATE FLAG
3102 1892         ;
3103 1892 4C00 A        LI      AC0,C0
3104 1893 B086 A        ST      AC0,@ATFST      ; RESET 'WAIT FOR SYNCH START' FLAG
3105 1894         ;
3106 1894 9483 A        LD      AC1,@ATCDS      ; TREAT PANEL UPDATE AS A DIAL CHANGE
3107 1895 211E A        JMP     TDCH
3108 1896         TPUE:
3109 1896         ;
3110 1896                                       ; DETERMINE CURRENT DIAL SELECTION
3111 1896 90A4 A TDIAL: LD      AC0,@AED23
3112 1897 0754 A        SKBIT   D4
3113 1898 2113 A        JMP     TND2                    ; NOT DIAL 2
3114 1899 0755 A        SKBIT   D5
3115 189A 210F A        JMP     TD2                     ; SELECT DIAL 2
3116 189B         ;                             DIAL SELECT ERROR
3117 189B 889C A        LD      AC2,ATERR               ; IF BOTH BITS 4 AND 5 ARE SET, RECORD
3118 189C                                               ; ERROR
3119 189C A202 A        ST      AC0,D2(AC2)             ; STORE IN ERROR BUFFER TERRO
3120 189D 8201 A        LD      AC0,D1(AC2)             ; FETCH ERROR COUNTER AND INCREMENT
3121 189E 4801 A        AISZ    AC0,C1
3122 189F 2100 A        JMP     .+1
3123 18A0 A201 A        ST      AC0,D1(AC2)
3124 18A1 8200 A        LD      AC0,D0(AC2)
3125 18A2 0720 A        SETBIT  D0                      ; SET ERROR FLAG BIT 0
3126 18A3 A200 A        ST      AC0,D0(AC2)
3127 18A4 9483 A        LD      AC1,@ATCDS              ; IN CASE OF ERROR, RETAIN CURRENT DIAL
3128 18A5 E588 A        SKG     AC1,T1                  ; SKIP IF NOT DIAL 1
3129 18A6 2109 A        JMP     TD1
3130 18A7 E587 A        SKG     AC1,T2                  ; SKIP IF DIAL 3
3131 18A8 2101 A        JMP     TD2
3132 18A9 2104 A        JMP     TD3
3133 18AA 4D02 A TD2:   LI      AC1,C2                  ; SELECT DIAL 2
3134 18AB 2105 A        JMP     TDSEL
3135 18AC 0755 A TND2:  SKBIT   D5                      ; SKIP IF DIAL 3
3136 18AD 2102 A        JMP     TD1
3137 18AE 4D03 A TD3:   LI      AC1,C3                  ; SELECT DIAL 3
3138 18AF 2101 A        JMP     TDSEL
3139 18B0 4D01 A TD1:   LI      AC1,C1                  ; SELECT DIAL 1
3140 18B1         ;
3141 18B1         TDSEL:
```

TIMING PLAN CPC - INITIATE NEXT CYCLE

```
18B1 8C83 A        LD      AC3,ATCDS          ; AC3=LOC OF CURRENT DIAL SELECTOR
18B2 F700 A        SKNE    AC1,(AC3)          ; SKIP IF NEW DIAL SELECTION
18B3 210C A        JMP     TNDC               ; JUMP NO DIAL CHANGE
18B4          ;                             DIAL CHANGE
18B4 B483 A TDCH:  ST      AC1,@ATCDS         ; CHANGE CURRENT DIAL SELECTOR
18B5 9C83 A        LD      AC3,@ATCDS         ; UPDATED DIAL SELECTION (1, 2, OR 3)
18B6 CC8A A        ADD     AC3,ATPP           ; ADDRESS PRECEDING POINTERS
18B7 8B00 A        LD      AC2,D0(AC3)        ; POINTER TO MASTER FOR CURRENT DIAL
18B8 8C8F A        LD      AC3,ATCTP          ; AC3=LOC OF CURRENT TP TABLE
18B9 2C36 A        JSR     @ATCOPY            ; COPY MASTER INTO CURRENT
18BA 9C83 A        LD      AC3,@ATCDS         ; UPDATED DIAL SELECTION (1, 2, OR 3)
18BB CC8A A        ADD     AC3,ATPP           ; ADDRESS PRECEDING POINTERS
18BC 8B00 A        LD      AC2,D0(AC3)        ; POINTER TO MASTER FOR CURRENT DIAL
18BD 8C8E A        LD      AC3,ATNTP          ; AC3=LOC OF NEXT TP
18BE 2C36 A        JSR     @ATCOPY            ;COPY MASTER TO NEXT
18BF 0200 A        RTS     D0                 ; RETURN FROM TNCY
18C0          ;
18C0          ;                             NO DIAL CHANGE
18C0 8C8F A TNDC:  LD      AC3,ATCTP
18C1 888E A        LD      AC2,ATNTP
18C2 2C36 A        JSR     @ATCOPY            ; COPY NEXT TP INTO CURRENT TP
18C3 9C83 A        LD      AC3,@ATCDS         ; UPDATED DIAL SELECTION (1, 2, OR 3)
18C4 CC8A A        ADD     AC3,ATPP           ; ADDRESS PRECEDING POINTERS
18C5 8B00 A        LD      AC2,D0(AC3)        ; POINTER TO MASTER FOR CURRENT DIAL
18C6 8C8E A        LD      AC3,ATNTP
18C7 2C36 A        JSR     @ATCOPY            ;COPY MASTER TO NEXT
18C8 0200 A        RTS     D0                 ; RETURN FROM TNCY
18C9          ;
```

```
3172 18C9                .LIST   LTP
3173 18C9        ;*****************************************************************************
3174 18C9        ;
3175 18C9        ;                                  TCOPY COPIES ONE 20 ELEMENT ARRAY INTO
3176 18C9        ;                                  ANOTHER 20 ELEMENT ARRAY
3177 18C9        ;                                  AC3 PASSES THE LOCATION OF ARRAY TO BE
3178 18C9        ;                                  CHANGED.  AC2 PASSES THE LOCATION OF
3179 18C9        ;                                  THE ARRAY TO BE COPIED
3180 18C9        ;*****************************************************************************
3181 18C9 4DEC A TCOPY:  LI      AC1,CM20
3182 18CA 8200 A TLOOP:  LD      AC0,(AC2)              ;VALUE OF THE OLD ARRAY
3183 18CB A300 A         ST      AC0,(AC3)              ; LOC AC3 IN NEW TABLE = AC0
3184 18CC 4901 A         AISZ    AC1,C1
3185 18CD 2101 A         JMP     .+2
3186 18CE 0200 A         RTS     D0                 ; RETURN FROM TCOPY
3187 18CF 4A01 A         AISZ    AC2,C1
3188 18D0 4B01 A         AISZ    AC3,C1
3189 18D1 21F8 A         JMP     TLOOP
3190 18D2 0200 A         RTS     D0                 ; RETURN FROM TCOPY
3191 18D3        ;
```

```
18D3                .LIST  LTP
18D3          ;
18D3          ;                    ... T I M I N G   P L A N   T A B L E S ...
18D3          ;
18D3          ;                        (ALL DATA IN THIS LOCAL DATA BASE SHALL BE
18D3          ;                         CONSIDERED AS APPLICATION DEPENDENT)
18D3          ;
18D3          ;                        PASSIVE BUS DETECTOR SYSTEM - PBD33J
18D3          ;
18D3          ;      (INTERVAL INDEX PARAMETERS ARE IN TERMS OF DISPLACEMENT FROM
18D3          ;       THE BEGINNING OF THE TIMING PLAN INTERVAL TABLE, 0 - N)
18D3          ;
18D3 0006 A TSCI:   .WORD   6              ; INDEX OF SYNCH CONTRACTIBLE INTERVAL
18D4 0004 A THI:    .WORD   4              ; INDEX OF TRANSFER/HOLD INTERVAL
18D5 0001 A TAI1:   .WORD   1              ; INDEX OF ADJ INT 1 (MUST BE IN MSG)
18D6 0006 A TAI2:   .WORD   6              ; INDEX OF ADJ INT 2
18D7 000A A TIMX:   .WORD   10             ; MAXIMUM INTERVAL INDEX
18D8          ;
18D8          ;                    ... PARAMETERS FOR EACH DETECTORIZED APPROACH ...
18D8          ;                          (MAXIMUM OF 16 APPROACHES)
18D8          TECI:
18D8          ;
18D8          ;                                33RD AND JOHNSON SOUTHBOUND APPROACH
18D8          ;
18D8 0006 A         .WORD   6                  ; PRIMARY CONTRACTIBLE INTERVAL
18D9 0001 A         .WORD   1                  ; SECONDARY CONTRACTIBLE INTERVAL
18DA 0003 A         .WORD   3                  ; PRIMARY EXTENDIBLE INTERVAL
18DB 0000 A         .WORD   0                  ; MINIMUM CONTRACTIBLE INTERVAL LENGTH
18DC          ;
18DC          ;                                33RD AND JOHNSON NORTHBOUND APPROACH
18DC          ;
18DC 0006 A         .WORD   6                  ; PRIMARY CONTRACTIBLE INTERVAL
18DD 0001 A         .WORD   1                  ; SECONDARY CONTRACTIBLE INTERVAL
18DE 0003 A         .WORD   3                  ; PRIMARY EXTENDIBLE INTERVAL
18DF 0000 A         .WORD   0                  ; MINIMUM CONTRACTIBLE INTERVAL LENGTH
18E0 1918 A         .=.+56
1918          ;
1918          ;                    ... SIGNAL DISPLAY COMMAND INDEX WORDS ...
1918          ;                          (MAXIMUM OF 20 INTERVALS)
1918          ;
1918          ;                        (BITS 15 - 8, MAIN STREET COMMAND INDEX
1918          ;                         BITS 7 - 0, CROSS STREET COMMAND INDEX)
1918          TCIW:
1918 0201 A         .WORD   X'0201             ; INTERVAL 0
1919 0201 A         .WORD   X'0201             ; INTERVAL 1
191A 0301 A         .WORD   X'0301             ; INTERVAL 2
191B 0401 A         .WORD   X'0401             ; INTERVAL 3
191C 0501 A         .WORD   X'0501             ; INTERVAL 4
191D 0102 A         .WORD   X'0102             ; INTERVAL 5
191F 0102 A         .WORD   X'0102             ; INTERVAL 6
191F 0103 A         .WORD   X'0103             ; INTERVAL 7
1920 0104 A         .WORD   X'0104             ; INTERVAL 8
```

```
3246 1921 0105 A        .WORD   X'0105          ; INTERVAL 9
3247 1922 192C A        .=.+10
3248 192C         ;
3249 192C         ;                    ... SKELETON TIMING PLAN TABLES ...
3250 192C         ;                          ( ADJ INTS = 0)
3251 192C         ;
3252 192C       TPTAB:
3253 192C 0046 A        .WORD   70                      ; TPM1
3254 192D 0000 A        .WORD   0               ; ADJ INT 1
3255 192F 0050 A        .WORD   80
3256 192F 0000 A        .WORD   0
3257 1930 001E A        .WORD   30
3258 1931 0046 A        .WORD   70
3259 1932 0000 A        .WORD   0               ; ADJ INT 2
3260 1933 006E A        .WORD   110
3261 1934 0000 A        .WORD   0
3262 1935 001E A        .WORD   30
3263 1936 FFFF A        .WORD   -1
3264 1937 1940 A        .=.+9
3265 1940         ;
3266 1940 0046 A        .WORD   70                      ; TPM2
3267 1941 0000 A        .WORD   0               ; ADJ INT 1
3268 1942 0050 A        .WORD   80
3269 1943 0000 A        .WORD   0
3270 1944 001E A        .WORD   30
3271 1945 0046 A        .WORD   70
3272 1946 0000 A        .WORD   0               ; ADJ INT 2
3273 1947 006E A        .WORD   110
3274 1948 0000 A        .WORD   0
3275 1949 001E A        .WORD   30
3276 194A FFFF A        .WORD   -1
3277 194B 1954 A        .=.+9
3278 1954         ;
3279 1954 0046 A        .WORD   70                      ; TPM3
3280 1955 0000 A        .WORD   0               ; ADJ INT 1
3281 1956 0050 A        .WORD   80
3282 1957 0000 A        .WORD   0
3283 1958 001E A        .WORD   30
3284 1959 0046 A        .WORD   70
3285 195A 0000 A        .WORD   0               ; ADJ INT 2
3286 195B 006E A        .WORD   110
3287 195C 0000 A        .WORD   0
3288 195D 001E A        .WORD   30
3289 195E FFFF A        .WORD   -1
3290 195F 1967 A        .=.+8
3291 1967 1968 A TPE:   .=.+1           ; TPE MUST BE LAST LOCATION IN TPM3
3292 1968         ; END OF TP TABLES
```

```
3293 1968                .PAGE
3294 1968                .ASECT
3295 1968 1A00 A         .=X'1A00
```

```
1A00          .LIST LPR
1A00        ;
1A00        ;                    ... P R E E M P T   C P C ...
1A00        ;
1A00        ;                                  PREEMPT MONITOR
1A00        ;                                  MPLS FIELD TEST
1A00        PMON:
1A00 2100 A         JMP     .+1                ; DEBUG HALT MAY BE INSERTED HERE VIA COTF
1A01        ;
1A01 2100 A         JMP     .+1
1A02        ;
1A02        ;                                  DECREMENT TIMER FOR RECENT BUS
1A02        ;                                  CLASSIFICATION TO ZERO
1A02 907B A         LD      AC0,@AVCBT
1A03 1103 A         BOC     CCEZ,.+4
1A04 48FF A         AISZ    AC0,CM1
1A05 2100 A         JMP     .+1
1A06 B07B A         ST      AC0,@AVCBT
1A07        ;                                  DECREMENT MIN. DURATION FOR PREEMPT
1A07        ;                                  TIMER TO ZERO
1A07 9075 A         LD      AC0,@APMDT             ; LOAD TIMER
1A08 1104 A         BOC     CCEZ,PM01              ; BRANCH IF TIMER = 0
1A09 48FF A         AISZ    AC0,CM1                ; DECREMENT COUNTER
1A0A 2100 A         JMP     .+1
1A0B B075 A         ST      AC0,@APMDT
1A0C 2107 A         JMP     PM02
1A0D        ;                                  MINIMUM DURATION TIMER = 0
1A0D 906E A PM01:   LD      AC0,@APIP
1A0E 1105 A         BOC     CCEZ,PM02              ; BRANCH IF PREEMPT NOT IN PROGRESS
1A0F                                               ; PREEMPT IN PROGRESS
1A0F        ;                                  EXAMINE CLEARANCE DETECTOR FOR
1A0F        ;                                  PREEMPT APPROACH
1A0F B07E A         ST      AC0,@AVCDT
1A10 2C31 A         JSR     @AVCLAS                ; CALL VEHICLE CLASSIFIER
1A11 1502 A         BOC     CCNZ,PM02              ; BRANCH IF VEHICLE NOT BUS
1A12 2C33 A         JSR     @APTER                 ; CALL PREEMPT TERMINATION
1A13 2100 A         JMP     .+1                ; DEBUG HALT MAY BE INSERTED HERE VIA COTF
1A14        ;
1A14        ;                                  CALL VEHICLE CLASSIFIER FOR ALL UPSTREAM
1A14        ;                                  (PREEMPT) DETECTORS
1A14        PM02:
1A14 4C01 A         LI      AC0,C1                 ; INITIALIZE DETECTOR INDEX
1A15        PM03:
1A15 B07E A         ST      AC0,@AVCDT
1A16 2C31 A         JSR     @AVCLAS                ; CALL VEHICLE CLASSIFIER
1A17 1502 A         BOC     CCNZ,PM05              ; JUMP IF VEHICLE NON-BUS
1A18 2C32 A         JSR     @APREQ                 ; CALL PREEMPT REQUEST ROUTINE
1A19 2100 A         JMP     .+1                ; DEBUG HALT MAY BE INSERTED HERE VIA COTF
1A1A 907E A PM05:   LD      AC0,@AVCDT
1A1B 4802 A         AISZ    AC0,C2                 ; INCREMENT DETECTOR INDEX BY 2
1A1C        ;                                      (ODD INDICES)
1A1C 2100 A         JMP     .+1
```

```
1A1D E116 A          SKG     AC0,PMHD          ; SKIP IF ALL UPSTREAM DETECTORS
1A1E        ;                                    SIGNATURE BLOCKS HAVE BEEN EXAMINED
1A1E 21F6 A          JMP     PM03
1A1F        ;                                  CALL VEHICLE CLASSIFIER FOR ALL
1A1F        ;                                  DOWNSTREAM (CLEARANCE) DETECTORS
1A1F 4C02 A          LI      AC0,C2            ; INITIALIZE DETECTOR INDEX
1A20        PM04:
1A20 B07E A          ST      AC0,@AVCDT
1A21 2C31 A          JSR     @AVCLAS           ; CALL VEHICLE CLASSIFIER
1A22 907E A          LD      AC0,@AVCDT
1A23 4802 A          AISZ    AC0,C2            ; INCREMENT DETECTOR INDEX BY 2
1A24        ;                                    (EVEN INDICES)
1A24 2100 A          JMP     .+1
1A25 E10E A          SKG     AC0,PMHD          ; SKIP IF ALL UPSTREAM DETECTORS
1A26        ;                                    SIGNATURE BLOCKS HAVE BEEN EXAMINED
1A26 21F9 A          JMP     PM04
1A27        ;                                  TERMINATION OF PREEMPT AT END OF
1A27        ;                                  EXTENDED INTERVAL
1A27 906E A          LD      AC0,@APIP
1A28 110A A          BOC     CCEZ,PMRET        ; IF PREEMPT NOT IN PROGRESS, RETURN.
1A29 9073 A          LD      AC0,@APMIF
1A2A 1101 A          BOC     CCSZ,.+2
1A2B 1307 A          BOC     CCZN,PMRET        ; IF EI IS IN NEXT CYCLE,RETURN.
1A2C 9084 A          LD      AC0,@ATIN         ; AC0 = CURRENT INTERVAL INDEX
1A2D 8073 A          LD      AC2,APMIF         ; AC2 = POINTER TO PMIF ARRAY
1A2E E200 A          SKG     AC0,D0(AC2)       ; SKIP IF CURRENT INTERVAL INDEX
1A2F                                           ; > INTERVAL INDEX SPECIFIED BY PMIF
1A2F 2103 A          JMP     PMRET
1A30 4C00 A          LI      AC0,C0
1A31 8073 A          ST      AC0,@APMIF
1A32 B06E A          ST      AC0,@APIP         ; RESET PIP FLAG
1A33 0200 A PMRET:   RTS     D0
1A34        ;
1A34        ;                                  LOCAL DATA
1A34 0004 A PMHD:    .WORD   DETS
```

```
1A35                .LIST LPR
1A35          PSWCH:
1A35          ;                                   DEBOUNCE SDU DETECTOR SWITCHES
1A35          ;
1A35          ;                                   CHECKOUT ROUTINE FOR INTERFACE
1A35          ;                                   BETWEEN LAB INTERSECTION MODEL AND
1A35          ;                                   VEHICLE SIGNATURE DATA
1A35          ;
1A35 9068 A          LD      AC0,@ABBUF                ; AC0 = DDIN
1A36 8880 A          LD      AC2,AVCSDB
1A37          ;                                   DEBOUNCE DOWNSTREAM SWITCH
1A37 075C A          SKBIT   D12                       ; SKIP IF DOWNSTREAM SWITCH UNACTIVATED
1A38 2103 A          JMP     .+4
1A39          ;                                   SWITCH IS UNACTIVATED
1A39 4D00 A          LI      AC1,C0
1A3A A511 A          ST      AC1,PSWFC                 ; CLEAR FLAG
1A3B 2105 A          JMP     .+6                       ; CHECK UPSTREAM DETECTOR
1A3C          ;                                   SWITCH IS ACTIVATED
1A3C 4D01 A          LI      AC1,C1
1A3D E50E A          SKG     AC1,PSWFC                 ; SKIP IF FLAG IS CLEAR
1A3E 2102 A          JMP     .+3
1A3F A50C A          ST      AC1,PSWFC                 ; SET FLAG
1A40          ;                                   MODIFY NEXT INSTR TO ACCESS OTHER
1A40          ;                                   SIGNATURES IN VCSDB
1A40 A60F A          ST      AC1,D15(AC2)              ; SET FOS STATUS WORD FOR DOWNSTREAM
1A41          ;                                   DEBOUNCE UPSTREAM SWITCH
1A41 075D A          SKBIT   D13                       ; SKIP IF UPSTREAM SWITCH UNACTIVATED
1A42 2103 A          JMP     .+4
1A43          ;                                   SWITCH IS UNACTIVATED
1A43 4D00 A          LI      AC1,C0
1A44 A508 A          ST      AC1,PSWFD                 ; CLEAR FLAG
1A45 0200 A          RTS     D0                        ; RETURN TO PMON
1A46          ;                                   SWITCH IS ACTIVATED
1A46 4D01 A          LI      AC1,C1
1A47 E505 A          SKG     AC1,PSWFD                 ; SKIP IF FLAG IS CLEAR
1A48 0200 A          RTS     D0
1A49 A503 A          ST      AC1,PSWFD                 ; SET FLAG
1A4A          ;                                   MODIFY NEXT INSTR TO ACCESS OTHER
1A4A          ;                                   SIGNATURES IN VCSDB
1A4A A607 A          ST      AC1,D7(AC2)               ; SET FOS STATUS WORD FOR UPSTREAM
1A4B 0200 A          RTS     D0                        ; RETURN
1A4C          ;
1A4C          ;                                   LOCAL DATA
1A4C 0000 A PSWFC:   .WORD   0
1A4D 0000 A PSWFD:   .WORD   0
1A4E          ;
```

```
1A4E                .LIST   LPR
1A4E          ;
1A4E          PCIM:
1A4E 2100 A         JMP     .+1          ; DEBUG HALT MAY BE INSERTED HERE VIA JSR @ACOTF
1A4F          ;                            PCIM IS CALLED BY PTER TO EXPAND
1A4F          ;                            A TIMING PLAN INTERVAL
1A4F          ;                                 APTITE: THE INTERVAL TO BE
1A4F          ;                                         EXPANDED
1A4F          ;                                 APTAVT: TIME AVAILABLE TO EXPAND IT
1A4F          ;                                 APTPMT: FIRST LOCATION OF THE
1A4F          ;                                         MASTER TIMING PLAN FOR
1A4F          ;                                         THE CURRENT DIAL SETTING
1A4F          ;                                 (IF ZERO, NO COMPARISON WITH
1A4F          ;                                         MASTER TABLE IS MADE)
1A4F          ;
1A4F          ;                            STORE AVAILABLE TIME IN PTEM
1A4F 9078 A         LD      AC0,@APTAVT
1A50 3381 A         RCPY    AC0,AC3
1A51 B06F A         ST      AC0,@APTEM
1A52          ;                            GENERATE ADDR POINTER TO CI
1A52 9079 A         LD      AC0,@APTITE
1A53 1808 A         BOC     CCZN,PC1             ; BRANCH IF BIT 15 SET
1A54          ;                            INTERVAL IS IN CURRENT CYCLE
1A54 8884 A         LD      AC2,ATIN
1A55 E200 A         SKG     AC0,D0(AC2)          ; SKIP IF CI > CURRENT INTERVAL INDEX
1A56 F200 A         SKNE    AC0,D0(AC2)          ; SKIP IF CI < CURRENT INTERVAL INDEX
1A57 2101 A         JMP     .+2
1A58          ;                            THE INTERVAL HAS EXPIRED
1A58 0201 A         RTS     D1                   ; RETURN FROM PCIM, CONTINUE PTER
1A59          ;                            INTERVAL HAS NOT EXPIRED
1A59 3281 A         RCPY    AC0,AC2
1A5A C88F A         ADD     AC2,ATCTP            ; AC2 POINTS AT CI
1A5B 2103 A         JMP     PC15
1A5C          ;
1A5C          PC1:
1A5C          ;                            INTERVAL IS IN NEXT CYCLE
1A5C 073F A         CLRBIT  D15
1A5D 3281 A         RCPY    AC0,AC2
1A5E C88E A         ADD     AC2,ATNTP            ; AC2 POINTS AT CI
1A5F          ;
1A5F          PC15:
1A5F 8600 A         LD      AC1,D0(AC2)          ; AC1 = DURATION OF CI
1A60 3D00 A         RADD    AC3,AC1              ; AC1 = DURATION CI + AVAILIBLE TIME
1A61          ;
1A61          ;                            SHOULD NEW DURATION OF CI BE COMPARED
1A61          ;                            WITH MASTER TIMING PLAN TABLE
1A61 907A A         LD      AC0,@APTPMT
1A62 1110 A         BOC     CCEZ,PC2
1A63 2100 A         JMP     .+1          ; DEBUG HALT MAY BE INSERTED HERE VIA JSR @ACOTF
1A64          ;                            COMPARE NEW DURATION WITH MASTER
1A64 9079 A         LD      AC0,@APTITE          ; AC0 = CI INDEX
1A65 073F A         CLRBIT  D15
```

```
1A66 8C7A A          LD      AC3,APTPMT        ; AC3 POINTS TO MASTER TIMING PLAN
1A67 C300 A          ADD     AC0,D0(AC3)       ; AC0 POINTS AT CI(MASTER)
1A68 3381 A          RCPY    AC0,AC3
1A69 E700 A          SKG     AC1,D0(AC3)       ; SKIP IF NEW DURATION > CI(MASTER)
1A6A 2108 A          JMP     PC2
1A6B          ;                               REDUCE NEW DURATION OF CI TO
1A6B          ;                               MAX(DURATION CI(MASTER),OLD DURATION CI)
1A6B 8200 A          LD      AC0,D0(AC2)       ; AC0 = DURATION OF CI
1A6C E300 A          SKG     AC0,D0(AC3)       ; SKIP IF AC0 > DURATION CI(MASTER)
1A6D 2101 A          JMP     .+2
1A6E          ;                               DURATION OF CI REMAINS THE SAME
1A6F 0201 A          RTS     D1                ;RETURN FROM PCIM, CONTINUE PTER
1A6F          ;                               EXPAND DURATION CI TO DURATION CI(MASTER)
1A6F D700 A          SUB     AC1,D0(AC3)       ; AC1 = AC1 - DURATION CI(MASTER)
1A70 B478 A          ST      AC1,@APTAVT
1A71 8700 A          LD      AC1,D0(AC3)       ; AC1 = DURATION CI(MASTER)
1A72 2102 A          JMP     .+3
1A73          PC2:
1A73 4C00 A          LI      AC0,C0
1A74 B078 A          ST      AC0,@APTAVT
1A75 A600 A          ST      AC1,D0(AC2)       ; DURATION CI = AC1
1A76 2100 A          JMP     .+1             ; DEBUG HALT MAY BE INSERTED HERE VIA JSR @ACOTF
1A77          ;                               IF CI INDEX = CURRENT INTERVAL INDEX,
1A77          ;                               ADJUST CURRENT INTERVAL TIMER
1A77 9079 A          LD      AC0,@APTITE
1A78 180B A          BOC     CCZN,PC3
1A79 8884 A          LD      AC2,ATIN
1A7A F200 A          SKNE    AC0,D0(AC2)       ; SKIP IF CI INDEX NOT = CURRENT
1A7B          ;                                  INTERVAL INDEX
1A7B 2101 A          JMP     .+2
1A7C 2107 A          JMP     PC3               ; RETURN FROM PCIM
1A7D 2100 A          JMP     .+1             ; DEBUG HALT MAY BE INSERTED HERE VIA JSR @ACOTF
1A7E          ;                               ADJUST CURRENT INTERVAL TIMER
1A7E 906F A          LD      AC0,@APTEM
1A7F 8C78 A          LD      AC3,APTAVT
1A80 D300 A          SUB     AC0,D0(AC3)       ; AC0 = TIME ADDED TO CI
1A81 8885 A          LD      AC2,ATINT       ; ADDRESS OF CURRENT INTERVAL TIMER
1A82 C200 A          ADD     AC0,D0(AC2)
1A83 B085 A          ST      AC0,@ATINT      ; MODIFY CURRENT INTERVAL TIMER
1A84          ;
1A84          ;                               IF ALL AVAILABLE TIME HAS BEEN RESTORED
1A84          ;                               RETURN FROM PCIM AND RETURN FROM PTER,
1A84          ;                               OTHERWISE RETURN FROM PCIM AND CONTINUE PTER
1A84 9078 A PC3:     LD      AC0,@APTAVT
1A85 1101 A          BOC     CCEZ,.+2
1A86 0201 A          RTS     D1                ; RETURN FROM PCIM, CONTINUE PTER
1A87 0200 A          RTS     D0                ; RETURN FROM PCIM, RETURN FROM PTER
```

```
1A88                .LIST   LPR
1A88        ;                                   PREEMPT TERMINATION
1A88        PTER:
1A88 2100 A         JMP     .+1                 ; DEBUG HALT VIA JSR @ACOTF
1A89        ;                                   RECORD TIME OF PREEMPT TERMINATION
1A89 949B A         LD      AC1,@AETCY
1A8A 88BA A         LD      AC2,ACTST
1A8B A601 A         ST      AC1,D1(AC2)
1A8C        ;
1A8C        ;                                   TEST WAIT/CONTROL BIT
1A8C 9069 A         LD      AC0,@ABDIN          ; AC0 = BDIN
1A8D 075E A         SKBIT   D14                 ; SKIP IF WAIT/CONTROL FLAG = 1
1A8E 2101 A         JMP     .+2
1A8F 0200 A PTRET:  RTS     D0                  ; RETURN FROM PTER
1A90        ;                                   TEST PREEMPT IN PROGRESS FLAG
1A90 906E A         LD      AC0,@APIP           ; AC0 = PREEMPT IN PROGRESS FLAG
1A91 887E A         LD      AC2,AVCDT           ; AC2 POINTS AT DETECTOR INDEX
1A92 F200 A         SKNE    AC0,D0(AC2)
1A93 2101 A         JMP     .+2
1A94 21FA A         JMP     PTRET
1A95        ;                                   CAN EXTENDIBLE PERIOD BE REDUCED
1A95 9073 A         LD      AC0,@APMIF          ; AC0 = PMIF
1A96 1802 A         BOC     CCZN,PT2            ; JUMP IF BIT 15 = 1
1A97 888F A         LD      AC2,ATCTP           ; AC2 = POINTER TO CURRENT CYCLE PLAN
1A98 2102 A         JMP     PT3
1A99 073F A PT2:    CLRBIT  D15
1A9A 888E A         LD      AC2,ATNTP           ; AC2 = POINTER TO NEXT CYCLE PLAN
1A9B 3200 A PT3:    RADD    AC0,AC2             ; AC2 POINTS AT EI
1A9C B86F A         ST      AC2,@APTEM
1A9D 9883 A         LD      AC2,@ATCDS          ; AC2 = DIAL SETTING
1A9E C38A A         ADD     AC2,ATPP            ;
1A9F 8600 A         LD      AC1,D0(AC2)         ; AC1 = POINTER TO MASTER TIMING PLAN
1AA0 B47A A         ST      AC1,@APTPMT         ; PTPMT = POINTER TO MASTER TIMING PLAN
1AA1 3100 A         RADD    AC0,AC1             ; AC1 = POINTER TO EI(MASTER)
1AA2 986F A         LD      AC2,@APTEM          ; AC2 = POINTER TO EI
1AA3 8E00 A         LD      AC3,D0(AC2)         ; AC3 = DURATION OF EI
1AA4 3681 A         RCPY    AC1,AC2
1AA5 DE00 A         SUB     AC3,D0(AC2)         ; AC3 = DURATION EI - DURATION EI(MASTER)
1AA6 3C81 A         RCPY    AC3,AC0
1AA7 1834 A         BOC     CCZN,PTEND          ; JUMP IF AC0 < 0
1AA8 1133 A         BOC     CCEZ,PTEND          ; JUMP IF AC0 = 0
1AA9        ;                                   HAS EXTENDIBLE PERIOD EXPIRED
1AA9 9073 A         LD      AC0,@APMIF          ; AC0 = PMIF
1AAA 1817 A         BOC     CCZN,PT6            ; JUMP IF BIT 15 = 1
1AAB 8884 A         LD      AC2,ATIN
1AAC E200 A         SKG     AC0,D0(AC2)         ; SKIP IF EI(INDEX) > CURRENT INTERVAL IN
1AAD 2101 A         JMP     .+2
1AAE 2113 A         JMP     PT6
1AAF F200 A         SKNE    AC0,D0(AC2)         ; SKIP IF EI INDEX < CURRENT INTERVAL
1AB0 2101 A         JMP     PT4
1AB1 212A A         JMP     PTEND
1AB2        ;                                   REDUCE EI
```

```
1AB2          ;
1AB2          ;                                   EI INDEX = CURRENT INTERVAL INDEX
1AB2 8885 A PT4:    LD      AC2,ATINT             ; AC2 = POINTER TO TIMER
1AB3 EE00 A         SKG     AC3,D0(AC2)           ; SKIP IF DURATION(EI - EI(MASTER))>TIMER
1AB4 2108 A         JMP     PT5
1AB5 9C6F A         LD      AC3,@APTEM            ; AC3 = POINTER TO EI
1AB6 8300 A         LD      AC0,D0(AC3)           ; AC0 = DURATION OF EI
1AB7 D200 A         SUB     AC0,D0(AC2)           ; AC0 = DURATION EI - TIMER
1AB8 A300 A         ST      AC0,D0(AC3)           ; DURATION EI = DURATION EI - TIMER
1AB9 9C85 A         LD      AC3,@ATINT
1ABA 4C00 A         LI      AC0,C0
1ABB BC85 A         ST      AC0,@ATINT            ; TIMER = 0
1ABC 2109 A         JMP     PT7
1ABD 5301 A PT5:    CAI     AC3,C1                ; AC3 = -DURATION(EI - EI(MASTER))
1ABE 9885 A         LD      AC2,@ATINT            ; AC2 = TIMER
1ABF 3E00 A         RADD    AC3,AC2               ; AC2 = TIMER - DURATION(EI - EI(MASTER))
1AC0 9885 A         ST      AC2,@ATINT
1AC1 5301 A         CAI     AC3,C1                ; AC3 = DURATION(EI - EI(MASTER))
1AC2          ;                                   EI INDEX > CURRENT INTERVAL INDEX
1AC2 3681 A PT6:    RCPY    AC1,AC2               ; AC2 = POINTER TO EI MASTER
1AC3 8200 A         LD      AC0,D0(AC2)           ; AC0 = DURATION EI(MASTER)
1AC4 986F A         LD      AC2,@APTEM            ; AC2 = POINTER TO EI
1AC5 A200 A         ST      AC0,D0(AC2)           ; DURATION EI = DURATION EI(MASTER)
1AC6          PT7:
1AC6 BC78 A         ST      AC3,@APTAVT           ; PTAVT = DURATION(EI-EI(MASTER))
1AC7          ;                                   EXPAND CI(S)
1AC7 8873 A         LD      AC2,APMIF
1AC8 8202 A         LD      AC0,D2(AC2)
1AC9 1103 A         BOC     CCEZ,PT8              ; IF CI(S) UNSPECIFIED, JUMP TO
1ACA          ;                                     EXPAND CI(P)
1ACA B079 A         ST      AC0,@APTITE
1ACB 2C34 A         JSR     @APCIM
1ACC 210F A         JMP     PTEND                 ; RETURN FROM PTER
1ACD          ;                                   EXPAND CI(P)
1ACD 4C00 A PT8:    LI      AC0,C0
1ACE B07A A         ST      AC0,@APTPMT
1ACF 8873 A         LD      AC2,APMIF
1AD0 8201 A         LD      AC0,D1(AC2)           ; AC0 POINTS TO CI(P)
1AD1 B079 A         ST      AC0,@APTITE
1AD2 2C34 A         JSR     @APCIM
1AD3 2108 A         JMP     PTEND                 ; RETURN FROM PTER
1AD4          ;
1AD4          ;                                   EXPAND CI(P) NEXT CYCLE
1AD4 8873 A         LD      AC2,APMIF
1AD5 8201 A         LD      AC0,D1(AC2)           ; AC0 = CI(P) INDEX
1AD6 073F A         CLRBIT  D15
1AD7 C08E A         ADD     AC0,ATNTP             ; AC0 POINTS AT CI(P)
1AD8 3281 A         RCPY    AC0,AC2
1AD9 8200 A         LD      AC0,D0(AC2)           ; AC0 = DURATION OF CI(P)
1ADA 3C00 A         RADD    AC3,AC0
1ADB A200 A         ST      AC0,D0(AC2)           ; STORE NEW DURATION OF CI(P)
1ADC 4C00 A PTEND:  LI      AC0,C0
```

```
3640 1ADD B06E A        ST      AC0,@APIP
3641 1ADE 2100 A        JMP     .+1                 ; DEBUG HALT MAY BE INSERTED HERE VIA JSR @ACOTF
3642 1ADF 0200 A        RTS     D0                          ; RETURN FROM PTER
3643 1AE0       ;
```

```
3646 1AF0                  .LIST   LPR
3647 1AF0 1AF0 A PMAS:     .WORD   PMAS
3648 1AF1 0001 A           .WORD   X'0001
3649 1AF2 0010 A           .WORD   X'0010
3650 1AF3 0100 A           .WORD   X'0100
3651 1AF4 0004 A PRHD:     .WORD   DETS                    ; HIGHEST DETECTOR INDEX EXPECTED
3652 1AF5 0000 A PRLD:     .WORD   0                       ; LOWEST DETECTOR INDEX EXPECTED - 1
```

```
1AE6            .LIST   LPR
1AE6        ;
1AE6        ;                                   LOCAL DATA
1AE6        ;
1AE6        PREO:
1AE6 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1AE7        ;                                   RECORD TIME OF PREEMPT REQUEST
1AE7 949B A         LD      AC1,@AETCY
1AE8 88BA A         LD      AC2,ACTST
1AE9 A600 A         ST      AC1,D0(AC2)
1AEA        ;                                   TEST WAIT/CONTROL FLAG
1AEA 9069 A         LD      AC0,@ABDIN              ; AC0 = BDIN
1AEB 075E A         SKBIT   D14                     ; SKIP IF WAIT/CONTROL FLAG IS SET
1AEC 2101 A         JMP     .+2
1AED 0200 A PRET:   RTS     D0                      ; RETURN FROM PREO
1AEE        ;                                   TEST PANEL SWITCH SETTING
1AEE 9883 A         LD      AC2,@ATCDS
1AEF C9F0 A         ADD     AC2,PMAS
1AF0 94BB A         LD      AC1,@ABPPD
1AF1 7600 A         SKAZ    AC1,D0(AC2)             ; SKIP IF SWITCH OFF
1AF2 2103 A         JMP     PRO
1AF3 0200 A         RTS     D0                      ; RETURN
1AF4        ;                                   ERROR HANDLING ROUTINES
1AF4 8C77 A PRER8:  LD      AC3,APERR
1AF5 947E A         LD      AC1,@AVCDT
1AF6 2C51 A         JSR     @AERE8
1AF7 0200 A         RTS     D0                      ; RETURN
1AF8 8C77 A PRER9:  LD      AC3,APERR
1AF9 947E A         LD      AC1,@AVCDT
1AFA 2C52 A         JSR     @AERE9
1AFB 4C00 A         LI      AC0,C0
1AFC BC6E A         ST      AC0,@APIP
1AFD 0200 A         RTS     D0                      ; RETURN
1AFE        ;
1AFE        ;                                   TEST PREEMPT IN PROGRESS FLAG
1AFE 906E A PRO:    LD      AC0,@APIP
1AFF 15ED A         BOC     CCNZ,PRET               ; BRANCH IF PIP FLAG SET
1B00        ;
1B00        ;                                   SET PREEMPT IN PROGRESS FLAG
1B00 907E A         LD      AC0,@AVCDT
1B01 E1E2 A         SKG     AC0,PRHD                ; SKIP IF VCDT > HIGHEST EXPECTED INDEX
1B02 E1E2 A         SKG     AC0,PRLD                ; SKIP IF VCDT > LOWEST EXPECTED INDEX
1B03 21F0 A         JMP     PRER8
1B04 3381 A         RCPY    AC0,AC3
1B05 4B01 A         AISZ    AC3,C1
1B06 BC6E A         ST      AC3,@APIP
1B07        ;                                   ACCESS APPROACH PARAMETERS
1B07 8C96 A         LD      AC3,ATECI
1B08 4BFF A         AISZ    AC0,CM1
1B09 2101 A         JMP     .+2
1B0A 2105 A         JMP     .+6
1B0B 1BEC A PR1:    BOC     CCZN,PRER9              ; BRANCH IF DOWNSTREAM DETECTOR
```

```
1B0C 4B04 A        AISZ    AC3,C4
1B0D 3081 A        NOP
1B0E 48FE A        AISZ    AC0,CM2
1B0F 21FB A        JMP     PR1
1B10        ;                                  DETERMINE 'MAXIMUM EXTENSION' FOR
1B10        ;                                  CURRENT DIAL SETTING
1B10 9883 A        LD      AC2,@ATCDS             ; AC2 = DIAL SETTING
1B11 C876 A        ADD     AC2,APEMAX
1B12 8200 A        LD      AC0,D0(AC2)            ; AC0 = E(MAX) FOR CURRENT
1B13        ;                                       DIAL SETTING
1B13 B076 A        ST      AC0,@APEMAX            ; PEMAX = AC0
1B14        ;
1B14        ;                                  DOES EI PRECEDE CI.
1B14 8800 A        LD      AC2,D0(AC3)            ; AC2 = CI
1B15 E302 A        SKG     AC2,D2(AC3)            ; SKIP IF CI > EI
1B16 214E A        JMP     P5                     ; JUMP TO PART D
1B17        ;                                  IF EI HAS NOT EXPIRED, JUMP TO PART C.
1B17 9484 A        LD      AC1,@ATIN              ; AC1 = CURRENT INTERVAL INDEX
1B18 E702 A        SKG     AC1,D2(AC3)            ; SKIP IF CII > EI
1B19 213B A        JMP     P3                     ; JUMP TO PART C
1B1A        ;                                  IF CI HAS EXPIRED, JUMP TO PART B.
1B1A F700 A        SKNE    AC1,D0(AC3)            ; SKIP IF CII NOT = CI
1B1B 2103 A        JMP     .+4
1B1C F700 A        SKG     AC1,D0(AC3)            ; SKIP IF CII > CI
1B1D 2104 A        JMP     .+5
1B1E 2117 A        JMP     P2                     ; JUMP TO PART B.
1B1F 9085 A        LD      AC0,@ATINT             ; AC0 = TIMER
1B20 E303 A        SKG     AC0,D3(AC3)            ; SKIP IF TIMER > C(MIN)
1B21 2114 A        JMP     P2                     ; JUMP TO PART B
1B22        ;
```

```
1822          .LIST   LPR
1822     ;                                  CONTRACTIBLE INTERVAL HAS NOT EXPIRED.
1822 2100 A   JMP     .+1                        ; JSR @ACOTF CAN BE INSERTED HERE
1823 8873 A   LD      AC2,APMIF                  ;AC2 POINTS AT PMIF ARRAY.
1824     ;                                  GENERATE POINTER TO SECONDARY CI
1824 808E A   LD      AC0,ATNTP
1825 C301 A   ADD     AC0,D1(AC3)
1826 8071 A   ST      AC0,@APTCS
1827 8301 A   LD      AC0,D1(AC3)
1828 072F A   SETBIT  D15
1829 A202 A   ST      AC0,D2(AC2)
182A     ;                                  GENERATE POINTER TO EI
182A 808E A   LD      AC0,ATNTP
182B C302 A   ADD     AC0,D2(AC3)
182C 8072 A   ST      AC0,@APTEI
182D 8302 A   LD      AC0,D2(AC3)
182E 072F A   SETBIT  D15
182F A200 A   ST      AC0,D0(AC2)
1830     ;                                  GENERATE POINTER TO CI
1830 8300 A   LD      AC0,D0(AC3)
1831 A201 A   ST      AC0,D1(AC2)
1832 888F A   LD      AC2,ATCTP
1833 3200 A   RADD    AC0,AC2
1834 8870 A   ST      AC2,@APTCP
1835 2175 A   JMP     P8
1836     ;
```

```
1B36          .LIST   LPR
1B36      P2:
1B36 2100 A     JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1B37 8873 A     LD      AC2,APMIF               ; AC2 POINTS AT PMIF ARRAY
1B38 4C00 A     LI      AC0,C0
1B39 A202 A     ST      AC0,D2(AC2)
1B3A B071 A     ST      AC0,@APTCS
1B3B      ;                                     GENERATE POINTER TO EI
1B3B 848E A     LD      AC1,ATNTP
1B3C 8302 A     LD      AC0,D2(AC3)
1B3D 3100 A     RADD    AC0,AC1
1B3E D472 A     ST      AC1,@APTEI
1B3F 072F A     SETBIT  D15
1B40 A200 A     ST      AC0,D0(AC2)
1B41      ;                                     GENERATE POINTER TO CI(P)
1B41 8300 A     LD      AC0,D0(AC3)
1B42 072F A     SETBIT  D15
1B43 A201 A     ST      AC0,D1(AC2)
1B44 8800 A     LD      AC2,D0(AC3)
1B45 C88E A     ADD     AC2,ATNTP
1B46 B870 A     ST      AC2,@APTCP              ; SAVE POINTER
1B47 8200 A     LD      AC0,(AC2)               ; AC0 = DURATION CI(P)
1B48 3181 A     RCPY    AC0,AC1                 ; AC1 = DURATION OF CI(P)
1B49 D303 A     SUB     AC0,D3(AC3)             ; AC0 = DURATION CI(P) - C(MIN)
1B4A 8876 A     LD      AC2,APEMAX
1B4B E200 A     SKG     AC0,D0(AC2)             ; SKIP IF DURATION(CI(P) - C(MIN))>E(MAX)
1B4C 2101 A     JMP     .+2
1B4D 8200 A     LD      AC0,D0(AC2)             ; AC0 = E(MAX)
1B4E      ;                                     REDUCE CI
1B4E 5001 A     CAI     AC0,C1
1B4F 3100 A     RADD    AC0,AC1                 ; AC1 = AC1 - AC0
1B50 5001 A     CAI     AC0,C1
1B51 9870 A     LD      AC2,@APTCP
1B52 A600 A     ST      AC1,D0(AC2)
1B53      ;                                     EXPAND EI
1B53 2500 A     JMP     @PT18A
1B54 1C04 A PT18A:.WORD P18A
1B55      ;
```

```
1855                .LIST   LPR
1855        P3:
1855 8873 A         LD      AC2,APMIF
1856 8300 A         LD      AC0,D0(AC3)         ; GENERATE POINTER TO CI(P)
1857 A201 A         ST      AC0,D1(AC2)
1858 C08F A         ADD     AC0,ATCTP
1859 B070 A         ST      AC0,@APTCP
185A 808E A         LD      AC0,ATNTP           ; GENERATE POINTER TO CI(S)
185B C301 A         ADD     AC0,D1(AC3)
185C B071 A         ST      AC0,@APTCS
185D 8301 A         LD      AC0,D1(AC3)
185E 072F A         SETBIT  D15
185F A202 A         ST      AC0,D2(AC2)
1860 8302 A         LD      AC0,D2(AC3)         ; GENERATE POINTER TO EI
1861 A200 A         ST      AC0,D0(AC2)
1862 C08F A         ADD     AC0,ATCTP
1863 B072 A         ST      AC0,@APTEI
1864 215F A         JMP     P10
1865        ;
```

```
1865                .LIST   LPR
1865        ;                                   HAS EXTENDIBLE INTERVAL EXPIRED
1865        P5:
1865 2100 A         JMP     .+1                 ; JSR @ACOTF CAN BE INSERTED HERE
1866 9084 A         LD      AC0,@ATIN
1867 E302 A         SKG     AC0,D2(AC3)         ; SKIP IF EI HAS EXPIRED.
1868 2115 A         JMP     P6                  ; JUMP TO PART E.
1869        ;                                   REDUCE DURATION OF CI TO MIN IN NEXT CYCLE.
1869        ;                                   ADD TIME FROM CI TO DURATION OF EI NEXT CYCLE.
1869 8873 A         LD      AC2,APMIF
186A 8300 A         LD      AC0,D0(AC3)
186B 072F A         SETBIT  D15
186C A201 A         ST      AC0,D1(AC2)
186D 8302 A         LD      AC0,D2(AC3)
186E 072F A         SETBIT  D15
186F A200 A         ST      AC0,D0(AC2)
1870 4C00 A         LI      AC0,C0
1871 A202 A         ST      AC0,D2(AC2)
1872 888E A         LD      AC2,ATNTP
1873 C800 A         ADD     AC2,D0(AC3)         ; AC2 POINTS TO CI(P)
1874 8703 A         LD      AC1,D3(AC3)         ; AC1 = C(MIN)
1875 8200 A         LD      AC0,(AC2)           ; AC2 = DURATION CI(P)
1876 A600 A         ST      AC1,(AC2)           ; DURATION CI(P) = C(MIN)
1877 D303 A         SUB     AC0,D3(AC3)         ; AC0 = DURATION CI(P) - C(MIN)
1878 888E A         LD      AC2,ATNTP
1879 C802 A         ADD     AC2,D2(AC3)         ; AC2 POINTS AT EI
187A C200 A         ADD     AC0,(AC2)           ; AC0 = AVAILABLE TIME + DURATION EI
187B A200 A         ST      AC0,(AC2)
187C 2500 A         JMP     @PT20
187D 1C10 A PT20:   .WORD   P20
187E        ;
```

```
1B7E                .LIST   LPR
1B7E          P6:
1B7E 2100 A         JMP     .+1                 ; JSR @ACOTF CAN BE INSERTED HERE
1B7F F300 A         SKNE    AC0,D0(AC3)         ; IF CI HAS NOT EXPIRED, JUMP TO PART F.
1B80 2103 A         JMP     .+4
1B81 E300 A         SKG     AC0,D0(AC3)         ; SKIP IF CURRENT INTERVAL > CI INDEX
1B82 2117 A         JMP     P7                  ; JUMP TO PART F
1B83 2104 A         JMP     .+5
1B84 9085 A         LD      AC0,@ATINT
1B85 E303 A         SKG     AC0,D3(AC3)         ; SKIP IF TIMER > C(MIN)
1B86 2101 A         JMP     .+2
1B87 2112 A         JMP     P7                  ; JUMP TO PART F.
1B88 8873 A         LD      AC2,APMIF           ; AC2 POINTS AT PMIF ARRAY
1B89          ;                               GENERATE POINTER TO CI(P)
1B89 808E A         LD      AC0,ATNTP
1B8A C300 A         ADD     AC0,D0(AC3)
1B8B B070 A         ST      AC0,@APTCP
1B8C 8300 A         LD      AC0,D0(AC3)
1B8D 072F A         SETBIT  D15
1B8E A201 A         ST      AC0,D1(AC2)
1B8F          ;                               GENERATE POINTER TO CI(S)
1B8F 808E A         LD      AC0,ATNTP
1B90 C301 A         ADD     AC0,D1(AC3)
1B91 B071 A         ST      AC0,@APTCS
1B92 8301 A         LD      AC0,D1(AC3)
1B93 072F A         SETBIT  D15
1B94 A201 A         ST      AC0,D1(AC2)
1B95          ;                               GENERATE POINTER TO EI
1B95 8302 A         LD      AC0,D2(AC3)
1B96 A200 A         ST      AC0,D0(AC2)
1B97 C08F A         ADD     AC0,ATCTP
1B98 B072 A         ST      AC0,@APTEI
1B99 212A A         JMP     P10
```

```
1B9A                .LIST   LPR
1B9A        P7:
1B9A 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1B9B 8873 A         LD      AC2,APMIF               ; AC2 POINTS AT PMIF ARRAY
1B9C        ;                                   GENERATE POINTERS TO CI(S)
1B9C 808E A         LD      AC0,ATNTP
1B9D C301 A         ADD     AC0,D1(AC3)
1B9E B071 A         ST      AC0,@APTCS
1B9F 8301 A         LD      AC0,D1(AC3)
1BA0 072F A         SETBIT  D15
1BA1 A202 A         ST      AC0,D2(AC2)
1BA2        ;                                   GENERATE POINTER TO EI
1BA2 8302 A         LD      AC0,D2(AC3)
1BA3 A200 A         ST      AC0,D0(AC2)
1BA4 C08F A         ADD     AC0,ATCTP
1BA5 B072 A         ST      AC0,@APTEI
1BA6        ;                                   GENERATE POINTER TO CI(P)
1BA6 8300 A         LD      AC0,D0(AC3)
1BA7 A201 A         ST      AC0,D1(AC2)
1BA8 888F A         LD      AC2,ATCTP
1BA9 3200 A         RADD    AC0,AC2
1BAA B870 A         ST      AC2,@APTCP
1BAB        P8:
1BAB 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1BAC 8200 A         LD      AC0,D0(AC2)             ; AC0 = DURATION OF CI(P)
1BAD 9484 A         LD      AC1,@ATIN
1BAE F700 A         SKNE    AC1,D0(AC3)             ; SKIP IF CURRENT INTERVAL NOT = CI INDEX
1BAF 2104 A         JMP     .+5
1BB0 D303 A         SUB     AC0,D3(AC3)             ; AC0 = DURATION OF CI(P) - C(MIN)
1BB1 8703 A         LD      AC1,D3(AC3)             ; AC1 = C(MIN)
1BB2 A600 A         ST      AC1,D0(AC2)             ; DURATION CI(P) = C(MIN)
1BB3 210B A         JMP     .+12
1BB4        ;                                   CI IS CURRENT CYCLE
1BB4 8885 A         LD      AC2,ATINT
1BB5 D200 A         SUB     AC0,D0(AC2)             ; AC0 = DURATION CI(P) - TIMER
1BB6 C303 A         ADD     AC0,D3(AC3)             ; AC0 = TIME LEFT
1BB7 9870 A         LD      AC2,@APTCP
1BB8 A200 A         ST      AC0,D0(AC2)
1BB9 9085 A         LD      AC0,@ATINT              ; AC0 = TIMER
1BBA D303 A         SUB     AC0,D3(AC3)             ; AC0 = TIMER - C(MIN)
1BBB 1201 A         BOC     CCZP,.+2
1BBC 4C00 A         LI      AC0,C0
1BBD        ;                                   AC0 = AVAILABLE TIME
1BBD 8703 A         LD      AC1,D3(AC3)
1BBE B485 A         ST      AC1,@ATINT              ; SET TIMER TO C(MIN)
1BBF 8876 A         LD      AC2,APEMAX
1BC0 E200 A         SKG     AC0,D0(AC2)             ; SKIP IF AC0 > E(MAX)
1BC1 F200 A         SKNE    AC0,D0(AC2)             ; SKIP IF AC0 NOT = E(MAX)
1BC2 2141 A         JMP     P18A
1BC3 212D A         JMP     P14A
1BC4        ;
```

```
1BC4                .LIST   LPR
1BC4          ;                                   ATTEMPT TO REDUCE CI(P)
1BC4          ;                                   CI(P) IS NOT THE CURRENT INTERVAL
1BC4          P10:
1BC4 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1BC5 9870 A         LD      AC2,@APTCP
1BC6 8200 A         LD      AC0,(AC2)               ; AC0 = DURATION CI(P)
1BC7 D303 A         SUB     AC0,D3(AC3)             ; AC0 = CI(P) - C(MIN)
1BC8 1B26 A         BOC     CCZN,P14                ; BRANCH TO P14 IF AC0 < ZERO
1BC9 8703 A         LD      AC1,D3(AC3)             ; AC1 = C(MIN)
1BCA A600 A         ST      AC1,(AC2)               ; DURATION OF CI(P) = C(MIN)
1BCB 8876 A         LD      AC2,APEMAX
1BCC F200 A         SKNE    AC0,D0(AC2)
1BCD 2134 A         JMP     P18
1BCE E200 A         SKG     AC0,D0(AC2)             ; SKIP IF AC0 > PEMAX
1BCF 2121 A         JMP     P14A
1BD0          ;                                   AVAILABLE TIME > E(MAX)
1BD0 D200 A         SUB     AC0,D0(AC2)             ; AC0 = DURATION CI(P) - C(MIN)
1BD1          ;                                           MAX DURATION EI
1BD1 3400 A         RADD    AC1,AC0                 ; AC0 = DURATION CI(P) - DURATION EI
1BD2 9870 A         LD      AC2,@APTCP
1BD3 A200 A         ST      AC0,(AC2)
1BD4 212D A         JMP     P18
1BD5          ;                                   CI(P) IS THE CURRENT INTERVAL
1BD5          P12:
1BD5 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1BD6 8085 A         LD      AC0,ATINT               ; AC0 = TIMER
1BD7 D303 A         SUB     AC0,D3(AC3)             ; AC0 = AVAILABLE TIME
1BD8 1B16 A         BOC     CCZN,P14                ; BRANCH TO P14 IF AC0 < ZERO
1BD9          ;                                   AVAILABLE TIME > ZERO
1BD9 9870 A         LD      AC2,@APTCP
1BDA 8600 A         LD      AC1,(AC2)               ; AC1 = DURATION OF CI(P)
1BDB 8885 A         LD      AC2,ATINT
1BDC D600 A         SUB     AC1,D0(AC2)             ; AC1 = DURATION CI(P) - TIMER
1BDD C703 A         ADD     AC1,D3(AC3)             ; AC1 = DURATION CI(P) - TIMER + C(MIN)
1BDE 9870 A         LD      AC2,@APTCP
1BDF A600 A         ST      AC1,(AC2)
1BE0 8903 A         LD      AC2,D3(AC3)             ; AC2 = C(MIN)
1BE1 8885 A         ST      AC2,@ATINT              ; SET TIMER TO C(MIN)
1BE2 8876 A         LD      AC2,APEMAX
1BE3 F200 A         SKNE    AC0,D0(AC2)             ; SKIP IF AVAILABLE TIME .NE. E(MAX)
1BE4 211D A         JMP     P18
1BE5 E200 A         SKG     AC0,D0(AC2)             ; SKIP IF AVAILABLE TIME > E(MAX)
1BE6 210A A         JMP     P14A
1BE7 D200 A         SUB     AC0,D0(AC2)             ; AC0 = AVAILABLE TIME - E(MAX)
1BE8 3100 A         RADD    AC0,AC1                 ; INCREASE DURATION OF CI(P)
1BE9 9870 A         LD      AC2,@APTCP
1BEA A200 A         ST      AC0,(AC2)               ; STORE NEW VALUE OF CI(P)
1BEB 8803 A         LD      AC2,D3(AC3)
1BEC 3200 A         RADD    AC0,AC2                 ; INCREASE TIMER
1BED 8885 A         ST      AC2,@ATINT              ; STORE NEW TIMER VALUE
1BEE 2113 A         JMP     P18
```

```
1BFF          P14:
1BFF 2100 A          JMP     .+1                  ; JSR @ACOTF CAN BE INSERTED HERE
1BF0 4C00 A          LI      AC0,0                ; AC0 = 0
1BF1 B06F A P14A:    ST      AC0,@APTEM           ; TEM HOLDS TIME AVAILABLE FROM CI(P)
1BF2 8876 A          LD      AC2,APEMAX
1BF3 D200 A          SUB     AC0,D0(AC2)          ; AC0 = TIME NEEDED FROM CI(S)
1BF4 9871 A          LD      AC2,@APTCS
1BF5 8600 A          LD      AC1,(AC2)            ; AC1 = DURATION CI(S)
1BF6 3400 A          RADD    AC1,AC0
1BF7 E303 A          SKG     AC0,D3(AC3)          ; SKIP IF AC0 > C(MIN)
1BF8 F303 A          SKNE    AC0,D3(AC3)          ; SKIP IF AC0 NOT = C(MIN)
1BF9 2106 A          JMP     .+7
1BFA          ;                                 TIME LEFT < C(MIN)
1BFA D703 A          SUB     AC1,D3(AC3)          ; AC1 = DURATION CI(S) - C(MIN)
1BFB 906F A          LD      AC0,@APTEM           ; AC0 = TIME AVAILABLE FROM CI(P)
1BFC 3400 A          RADD    AC1,AC0              ; AC0 = TOTAL TIME AVAILABLE
1BFD 8703 A          LD      AC1,D3(AC3)
1BFE A600 A          ST      AC1,(AC2)            ; DURATION CI(S) = C(MIN)
1BFF 2104 A          JMP     P18A
1C00          ;                                 TIME LEFT => C(MIN)
1C00 A200 A          ST      AC0,(AC2)
1C01 2100 A          JMP     P18
1C02          ;
1C02          ;                                 EXPAND EI.
1C02          P18:
1C02 2100 A          JMP     .+1                  ; JSR @ACOTF CAN BE INSERTED HERE
1C03 9076 A          LD      AC0,@APEMAX          ; AC0 = MAX EXTENSION
1C04          ;                                 AC0 = AVAILABLE TIME
1C04 9872 A P18A:    LD      AC2,@APTEI
1C05 1010 A          BOC     CCZN,P21             ; BRANCH IF NO TIME AVAILIBLE
1C06 8600 A          LD      AC1,(AC2)            ; AC1 = DURATION EI
1C07 3100 A          RADD    AC0,AC1              ; AC1 = AVAILABLE TIME + DURATION EI
1C08 A600 A          ST      AC1,(AC2)            ; DURATION EI = AC1
1C09 E88E A          SKG     AC2,ATNTP
1C0A 2105 A          JMP     .+6
1C0B 9485 A          LD      AC1,@ATINT           ; AC1 = TIMER
1C0C 3100 A          RADD    AC0,AC1              ; AC1 = ADJUSTED TIMER
1C0D 9084 A          LD      AC0,@ATIN            ; AC0 = INTERVAL INDEX
1C0E F302 A          SKNE    AC0,D2(AC3)
1C0F B485 A          ST      AC1,@ATINT
1C10 4C19 A P20:     LI      AC0,DPMD             ; SET MINIMUM DURATION TIMER
1C11 B075 A          ST      AC0,@APMDT
1C12 B074 A          ST      AC0,@APFREQ
1C13          ;
1C13          ;                                 RESET WAIT FOR SYNCH START FLAG
1C13 4C00 A          LI      AC0,C0
1C14 B086 A          ST      AC0,@ATFST
1C15          ;
1C15 0200 A          RTS     D0                   ; RETURN FROM PREQ
1C16          ;                                 UNABLE TO PROVIDE PREEMPT
1C16 4C00 A P21:     LI      AC0,C0
1C17 B06E A          ST      AC0,@APIP            ; CLEAR PIP FLAG
```

```
4053 1C18 0200 A      RTS     DO                    ; RETURN FROM PREQ
4054 1C19             ;
4055 1C19             ;
4056 1C19             ;
```

```
1C19                .LIST   LVC
1C19        VCLAS:
1C19        ;                                   VEHICLE CLASSIFICATION
1C19        ;                                   MAY 30 1975
1C19        ;
1C19 2100 A         JMP     .+1                     ; JSR @ACOTF MAY BE INSERTED HERE
1C1A 9C7E A         LD      AC3,@AVCDT
1C1B        ;                                   IS THE SPECIFIED DETECTOR IN RANGE
1C1B ED66 A         SKG     AC3,VCHD
1C1C ED66 A         SKG     AC3,VCLD
1C1D 2154 A         JMP     VCER1
1C1E        ;                                   LOCATE THIS DETECTOR'S PORTION OF SDB
1C1E 48FF A         AISZ    AC3,CM1
1C1F 2100 A         JMP     .+1
1C20 5F03 A         SHL     AC3,D3
1C21 8080 A         LD      AC2,AVCSDB
1C22 3F00 A         RADD    AC2,AC3
1C23 887F A         LD      AC2,AVCDB
1C24        ;                                   CHECK EOS STATUS REGISTER
1C24 8307 A         LD      AC0,D7(AC3)
1C25 1118 A         BOC     CCEZ,VCRET
1C26        ;
1C26        ;                                   ALL DATA FROM THE VEHICLE SIGNATURE DATA
1C26        ;                                   BASE IS SCALED BO
1C26        ;
1C26        ;                                   COPY VEHICLE SIGNATURE DATA BASE
1C26 8305 A         LD      AC0,D5(AC3)
1C27 A205 A         ST      AC0,D5(AC2)
1C28 8300 A         LD      AC0,D0(AC3)
1C29 A200 A         ST      AC0,D0(AC2)
1C2A 8301 A         LD      AC0,D1(AC3)
1C2B A201 A         ST      AC0,D1(AC2)
1C2C 8302 A         LD      AC0,D2(AC3)
1C2D A202 A         ST      AC0,D2(AC2)
1C2E 8303 A         LD      AC0,D3(AC3)
1C2F A203 A         ST      AC0,D3(AC2)
1C30 8304 A         LD      AC0,D4(AC3)
1C31 A204 A         ST      AC0,D4(AC2)
1C32 8306 A         LD      AC0,D6(AC3)
1C33 A206 A         ST      AC0,D6(AC2)
1C34        ;                                   RECHECK EOS STATUS REGISTER
1C34 8307 A         LD      AC0,D7(AC3)
1C35        ;                                   RESET EOS STATUS REGISTER
1C35 4D00 A         LI      AC1,C0
1C36 A707 A         ST      AC1,D7(AC3)
1C37 1508 A         BOC     CCNZ,VC01
1C38 947E A         LD      AC1,@AVCDT
1C39 8C77 A         LD      AC3,APERR
1C3A 2C4F A         JSR     @AERE6
1C3B 8205 A         LD      AC0,D5(AC2)
1C3C 1503 A         BOC     CCNZ,VC01
1C3D 2C50 A         JSR     @AERE7
```

```
1C3E 4C01 A VCRET:  LI    AC0,C1
1C3F 0200 A         RTS   D0                    ; RETURN FROM VCLAS
1C40        ;                                 TEST FOR NOISE
1C40 4C20 A VC01;   LI    AC0,32
1C41 E200 A         SKG   AC0,D0(AC2)
1C42 210A A         JMP   VC02
1C43        ;                                 TOTAL TIME OF SIGNATURE TOO SHORT FOR VEHICLE
1C43        ;                                 SIGNATURE ASSUMED TO BE NOISE
1C43 947E A         LD    AC1,@AVCDT
1C44 8C77 A         LD    AC3,APERR
1C45 2C4E A         JSR   @AERE5
1C46 4C01 A         LI    AC0,C1
1C47 8C7D A         LD    AC3,AVCLOG
1C48 7D01 A         ISZ   D1(AC3)
1C49 0200 A         RTS   D0                    ; RETURN FROM VCLAS
1C4A 7D00 A         ISZ   D0(AC3)
1C4B 3081 A         NOP
1C4C 0200 A         RTS   D0                    ; RETURN FROM VCLAS
1C4D        ;                                 INCREMENT DETECTOR COUNTER
1C4D        VC02:
1C4D 4FFF A         LI    AC3,CM1
1C4E 947E A         LD    AC1,@AVCDT
1C4F 3700 A         RADD  AC1,AC3
1C50 CC7C A         ADD   AC3,AVCVC
1C51 7D00 A         ISZ   D0(AC3)
1C52 2100 A         JMP   .+1
1C53        ;                                 GROSS TEST
1C53 4C00 A         LI    AC0,C0
1C54 F205 A         SKNE  AC0,D5(AC2)
1C55 2115 A         JMP   VCGTF
1C56        ;                                 CHECK MAGNITUDE OF SIGNATURE DATA
1C56 812D A         LD    AC0,VCMA
1C57 7201 A         SKAZ  AC0,D1(AC2)
1C58 211D A         JMP   VCER2
1C59 7203 A         SKAZ  AC0,D3(AC2)
1C5A 211B A         JMP   VCER2
1C5B 7205 A         SKAZ  AC0,D5(AC2)
1C5C 2119 A         JMP   VCER2
1C5D 8127 A         LD    AC0,VCMT
1C5E 7200 A         SKAZ  AC0,D0(AC2)
1C5F 211A A         JMP   VCER3
1C60 7202 A         SKAZ  AC0,D2(AC2)
1C61 2118 A         JMP   VCER3
1C62 7204 A         SKAZ  AC0,D4(AC2)
1C63 2116 A         JMP   VCER3
1C64 7206 A         SKAZ  AC0,D6(AC2)
1C65 2114 A         JMP   VCER3
1C66        ;                                 CHECK FOR TIME OVERFLOW
1C66 811F A         LD    AC0,VCMOF
1C67 6200 A         AND   AC0,D0(AC2)
1C68 F11D A         SKNE  AC0,VCMOF
1C69 2152 A         JMP   VCHFE
```

```
1C6A 2127 A          JMP    VCDHK
1C6B 4C01 A VCGTF:   LI     AC0,C1
1C6C 8C7D A          LD     AC3,AVCLOG
1C6D 7B03 A          ISZ    D3(AC3)
1C6E 0200 A          RTS    D0                    ; RETURN FROM VCLAS
1C6F 7B02 A          ISZ    D2(AC3)
1C70 2100 A          JMP    .+1
1C71 0200 A          RTS    D0                    ; RETURN FROM VCLAS
1C72         VCER1:
1C72         ;                                   DETECTOR SPECIFIED DOES NOT EXIST
1C72 947E A          LD     AC1,@AVCDT
1C73 8C77 A          LD     AC3,APERR
1C74 2C4A A          JSR    @AERE1
1C75 21C8 A          JMP    VCRET
1C76         VCER2:
1C76         ;                                   VEHICLE SIGNATURE DATA EXCEEDS BOUNDS (AMP)
1C76 947E A          LD     AC1,@AVCDT
1C77 8C77 A          LD     AC3,APERR
1C78 2C4B A          JSR    @AERE2
1C79 21C4 A          JMP    VCRET
1C7A         VCER3:
1C7A         ;                                   VEHICLE SIGNATURE DATA EXCEEDS BOUNDS (TIME)
1C7A 947E A          LD     AC1,@AVCDT
1C7B 8C77 A          LD     AC3,APERR
1C7C 2C4C A          JSR    @AERE3
1C7D 21C0 A          JMP    VCRET
1C7E         VCER4:
1C7E         ;                                   VEHICLE TIME DATA OUT OF ORDER
1C7E 8C77 A          LD     AC3,APERR
1C7F 947E A          LD     AC1,@AVCDT
1C80 2C4D A          JSR    @AERE4
1C81 21BC A          JMP    VCRET
```

```
1C82                    .LIST   LVC
1C82            ;                                   DETECTOR INDEX RANGE
1C82 0004 A VCHD:       .WORD   DETS                ; HIGHEST DETECTOR INDEX NUMBER EXPECTED
1C83 0000 A VCLD:       .WORD   0                   ; LOWEST DETECTOR INDEX NUMBER - 1
1C84            ;                                   MASKS
1C84 FF00 A VCMA:       .WORD   X'FF00              ; MASK USED IN AMPLITUDE IN RANGE TEST
1C85 F000 A VCMT:       .WORD   X'F000              ; MASK USED IN TIME IN RANGE TEST
1C86 0FFF A VCMOF:      .WORD   X'0FFF              ; MASK USED IN TIME OVERFLOW TEST
1C87            ;                                   THE FOLLOWING ARE USED IN VCDHK
1C87 0028 A VCST1:      .WORD   40                  ; 2.50 B4      USED IN L2 TEST
1C88 0007 A VCST2:      .WORD   7                   ; 0.43 B4      USED IN L1 TEST
1C89 000B A VCST3:      .WORD   11                  ; 0.70 B4      USED IN L1 TEST
1C8A            ;                                   THE FOLLOWING ARE USED IN VCHFE
1C8A 2BAE A VC31:       .WORD   11182               ; 2.730 B13    G1 COEFFICIENT FOR H1
1C8B 0F04 A VC32:       .WORD   3588                ; 0.876 B13    G2 COEFFICIENT FOR H1
1C8C 2851 A VC41:       .WORD   10321               ; 2.520 B13    G1 COEFFICIENT FOR H2
1C8D 0239 A VC42:       .WORD   569                 ; 0.139 B13    G2 COEFFICIENT FOR H2
1C8E 3EE1 A H1T1:       .WORD   16097               ; 3.93 B13     H1 TEST LOWER BOUND
1C8F 6785 A H1T2:       .WORD   26501               ; 6.47 B13     H1 TEST UPPER BOUND
1C90 3000 A H2T1:       .WORD   12288               ; 3.00 B13     H2 TEST LOWER BOUND
1C91 5000 A H2T2:       .WORD   20480               ; 5.00 B13     H2 TEST UPPER BOUND
```

```
1C92          .LIST   LVC
1C92      VCDHK:
1C92      ;                                   VEHICLE CLASSIFICATION  MOVING/STOPPED TEST
1C92      ;                                   INSURE TIME DATA WORDS IN PROPER ORDER
1C92 8204 A         LD      AC0,D4(AC2)
1C93 E202 A         SKG     AC0,D2(AC2)
1C94 21E9 A         JMP     VCER4
1C95 8206 A         LD      AC0,D6(AC2)
1C96 E204 A         SKG     AC0,D4(AC2)
1C97 21E6 A         JMP     VCER4
1C98 8200 A         LD      AC0,D0(AC2)
1C99 E206 A         SKG     AC0,D6(AC2)
1C9A 21E3 A         JMP     VCER4
1C9B      ;                                   FORM T(C1) + T(C4) - T(C3)
1C9B 8200 A         LD      AC0,D0(AC2)         ; AC0 = T(C4)
1C9C D206 A         SUB     AC0,D6(AC2)         ; AC0 = T(C4) - T(C3)
1C9D C202 A         ADD     AC0,D2(AC2)         ; AC0 = T(C4) - T(C3) + T(C1)
1C9E A207 A         ST      AC0,D7(AC2)         ; B0
1C9F      ;                                   FORM F0
1C9F 8602 A         LD      AC1,D2(AC2)         ; B0
1CA0 5D04 A         SHL     AC1,D4              ; B4
1CA1 4C00 A         LI      AC0,C0
1CA2 C172 A         ADD     AC0,VCC0            ; CLEAR OVERFLOW STATUS BIT
1CA3 0690 A         DIV     D7(AC2)             ; B4
1CA4 0007 A
1CA5 2100 A         JMP     .+1
1CA6      ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1CA6      ;                                   JUMP TO VCHFE
1CA6 074E A         SKSTF   D14
1CA7 2101 A         JMP     .+2
1CA8 2113 A         JMP     VCHFE
1CA9      ;                                   AC1 = T(C1) / T(C1) + T(C4) - T(C3)
1CA9      ;                                   VEHICLE IS CONSIDERED MOVING VEHICLE IF
1CA9      ;                                   0.43 < F0 < 0.7
1CA9 E5DE A         SKG     AC1,VCST2           ; SKIP IF AC1 > 0.43
1CAA 2111 A         JMP     VCHFE
1CAB E5DD A         SKG     AC1,VCST3           ; SKIP IF AC1 > 0.7
1CAC F5DC A         SKNE    AC1,VCST3           ; SKIP IF AC1 NOT = 0.7
1CAD 210E A         JMP     VCHFE               ; JUMP TO STOPPED VEHICLE CLASSIFIER
1CAE      ;                                   FORM F1 + F2
1CAE 8606 A         LD      AC1,D6(AC2)         ; B0
1CAF D602 A         SUB     AC1,D2(AC2)         ; AC1 = T(C3) - T(C1)
1CB0 5D04 A         SHL     AC1,D4              ; B4
1CB1 4C00 A         LI      AC0,C0
1CB2 C162 A         ADD     AC0,VCC0            ; CLEAR OVERFLOW STATUS BIT
1CB3 0690 A         DIV     D7(AC2)             ; B4
1CB4 0007 A
1CB5 2100 A         JMP     .+1
1CB6      ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1CB6      ;                                   JUMP TO VCHFE
1CB6 074E A         SKSTF   D14
```

```
4270 1CB7 2101 A        JMP     .+2
4271 1CB8 2103 A        JMP     VCHFE
4272 1CB9          ;                          AC1 = T(C3) - T(C1) / T(C1) + T(C4) - T(C3)
4273 1CB9          ;                          VEHICLE IS CONSIDERED STOPPED VEHICLE IF
4274 1CB9          ;                          F1 + F2 > 2.5
4275 1CB9 E5CD A        SKG     AC1,VCST1          ; SKIP IF AC1 > 2.5
4276 1CBA 2132 A        JMP     VCKFE              ; JUMP TO MOVING VEHICLE CLASSIFIER
4277 1CBB 2100 A        JMP     VCHFE
```

```
1CBC                .LIST  LVC
1CBC          ;                                VEHICLE CLASSIFICATION
1CBC          ;                                STOPPED VEHICLE CLASSIFIER
1CBC         VCHFE:
1CBC 8C7D A          LD     AC3,AVCLOG
1CBD 7B05 A          ISZ    D5(AC3)
1CBE 2102 A          JMP    .+3
1CBF 7B04 A          ISZ    D4(AC3)
1CC0 2100 A          JMP    .+1
1CC1          ;                                INCREMENT STOPPED VEHICLE COUNTER
1CC1          ;                                FORM C31 * A1
1CC1 85C8 A          LD     AC1,VC31                ; B13
1CC2 0680 A          MPY    D3(AC2)
     1CC3 0003 A
1CC4 A208 A          ST     AC0,D8(AC2)
1CC5 A609 A          ST     AC1,D9(AC2)
1CC6          ;                                FORM C32 * A2
1CC6 85C4 A          LD     AC1,VC32                ; B13
1CC7 0680 A          MPY    D5(AC2)
     1CC8 0005 A
1CC9          ;                                FORM (C31*A1 + C32*A2) / A0
1CC9 06A0 A          DADD   D8(AC2)                 ; B13
     1CCA 0008 A
1CCB C149 A          ADD    AC0,VCC0                ; CLEAR OVERFLOW STATUS BIT
1CCC 0690 A          DIV    D1(AC2)                 ; B13
     1CCD 0001 A
1CCE 2100 A          JMP    .+1
1CCF          ;                                CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1CCF          ;                                JUMP TO VCNBS
1CCF 074E A          SKSTF  D14
1CD0 2101 A          JMP    .+2
1CD1 2552 A          JMP    @AVCVNB
1CD2          ;                                IF VEHICLE IS NOT A BUS, JUMP TO VCVNB.
1CD2 E5BB A          SKG    AC1,H1T1                ; SKIP IF AC1 > 3.93
1CD3 2550 A          JMP    @AVCVNB
1CD4 E5BA A          SKG    AC1,H1T2                ; SKIP IF AC1 => 6.47
1CD5 2101 A          JMP    .+2
1CD6 254D A          JMP    @AVCVNB
1CD7          ;                                FORM C41 * A1
1CD7 85B4 A          LD     AC1,VC41                ; B13
1CD8 0680 A          MPY    D3(AC2)
     1CD9 0003 A
1CDA A208 A          ST     AC0,D8(AC2)
1CDB A609 A          ST     AC1,D9(AC2)
1CDC          ;                                FORM C42 * A2
1CDC 85B0 A          LD     AC1,VC42                ; B13
1CDD 0680 A          MPY    D5(AC2)
     1CDE 0005 A
1CDF          ;                                FORM (C41*A1 + C42*A2) / A0
1CDF 06A0 A          DADD   D8(AC2)                 ; B13
     1CE0 0008 A
1CE1 C133 A          ADD    AC0,VCC0                ; CLEAR OVERFLOW STATUS BIT
```

```
1CE2 0690 A        DIV     D1(AC2)          ; B13
1CE3 0001 A
1CE4 2100 A        JMP     .+1
1CE5          ;                             CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1CE6          ;                             JUMP TO VCNBS
1CE5 074E A        SKSTF   D14
1CE6 2101 A        JMP     .+2
1CE7 253C A        JMP     @AVCVNB
1CE8          ;                             IF VEHICLE IS NOT A BUS, JUMP TO VCVNB
1CE8 E5A7 A        SKG     AC1,H2T1         ; SKIP IF AC1 > 3.0
1CE9 253A A        JMP     @AVCVNB
1CEA E5A6 A        SKG     AC1,H2T2         ; SKIP IF AC1 > 5.0
1CEB 2537 A        JMP     @AVCBUS
1CEC 2537 A VCHE:  JMP     @AVCVNB
```

```
1CED                .LIST   LVC
1CED          ;                                   VEHICLE CLASSIFICATION
1CED          ;                                   MOVING VEHICLE CLASSIFIER
1CED          ;                                       BIT 13 = 1.0
1CED          VCKFE:
1CED 8C7D A           LD      AC3,AVCLOG
1CEE 7807 A           ISZ     D7(AC3)
1CEF 2102 A           JMP     .+3
1CF0 7806 A           ISZ     D6(AC3)
1CF1 2100 A           JMP     .+1
1CF2 8021 A           LD      AC3,VCCP
1CF3 4000 A           LI      AC0,C0
1CF4 A20C A           ST      AC0,D12(AC2)
1CF5 A20F A           ST      AC0,D15(AC2)
1CF6          VCK:
1CF6          ;                                   FORM C4*A2 = G2'
1CF6 8704 A           LD      AC1,D4(AC3)             ; B13
1CF7 0680 A           MPY     D5(AC2)
1CF8 0005 A
1CF9 A208 A           ST      AC0,D8(AC2)
1CFA A609 A           ST      AC1,D9(AC2)
1CFB          ;                                   FORM C3*A1 = G1'
1CFB 8703 A           LD      AC1,D3(AC3)             ; B13
1CFC 0680 A           MPY     D3(AC2)
1CFD 0003 A
1CFE A20A A           ST      AC0,D10(AC2)
1CFF A605 A           ST      AC1,D11(AC2)
1D00          ;                                   FORM ABS(G1' - G2') / A0
1D00 E208 A           SKNE    AC0,D8(AC2)             ; SKIP IF UPPER G1' NOT = UPPER G2'
1D01 2103 A           JMP     .+4
1D02 E208 A           SKG     AC0,D8(AC2)             ; SKIP IF UPPER G1' > UPPER G2'
1D03 212F A           JMP     VCK3
1D04 2122 A           JMP     VCK21
1D05 651C A           AND     AC1,VCM15
1D06 F51B A           SKNE    AC1,VCM15               ; SKIP IF BIT 15 IS NOT SET
1D07 2108 A           JMP     VCK1
1D08 8519 A           LD      AC1,VCM15
1D09 6609 A           AND     AC1,D9(AC2)
1D0A F517 A           SKNE    AC1,VCM15               ; SKIP IF BIT 15 IS NOT SET
1D0B 2127 A           JMP     VCK3
1D0C 860B A VCK0:     LD      AC1,D11(AC2)
1D0D E609 A           SKG     AC1,D9(AC2)             ; SKIP IF LOWER G1' > LOWER G2'
1D0E 2124 A           JMP     VCK3
1D0F 2116 A           JMP     VCK2
1D10 6609 A VCK1:     AND     AC1,D9(AC2)
1D11 F510 A           SKNE    AC1,VCM15
1D12 21F9 A           JMP     VCK0
1D13 2112 A           JMP     VCK2
```

```
1D14                .LIST   LVC
1D14 1D15 A VCCP:   .WORD   VCCO
1D15        ;                          THE FOLLOWING IS USED TO CLEAR THE OVERFLOW
1D15        ;                          STATUS BIT BEFORE EACH DIVIDE OPERATION
1D15 0000 A VCCO:   .WORD   0
1D16        ;                          THE FOLLOWING ARE USED IN VCKFE
1D16 046E A         .WORD   1134       ; 0.277 B13    F1 COEFFICIENT FOR K4
1D17 0051 A         .WORD   81         ; 0.0198 B13   F2 COEFFICIENT FOR K4
1D18 03A5 A         .WORD   933        ; 0.228 B13    G1 COEFFICIENT FOR K4
1D19 0F26 A         .WORD   3878       ; 0.947 B13    G2 COEFFICIENT FOR K4
1D1A 0000 A         .WORD   0
1D1B 1028 A         .WORD   4136       ; 1.01 B13     K4 BIAS
1D1C        ;
1D1C 1A8F A         .WORD   6799       ; 1.66 B13     F1 COEFFICIENT FOR K5
1D1D 15C2 A         .WORD   5570       ; 1.36 B13     F2 COEFFICIENT FOR K5
1D1E 22B8 A         .WORD   8888       ; 2.17 B13     G1 COEFFICIENT FOR K5
1D1F 4199 A         .WORD   16793      ; 4.10 B13     G2 COEFFICIENT FOR K5
1D20 0000 A         .WORD   0
1D21 1F1A A         .WORD   7962       ; 1.944 B13    K5 BIAS
1D22        ;                          MASK
1D22 8000 A VCM15:  .WORD   X'8000     ; MASK FOR DOUBLE PRECISION COMPARISIONS
1D23 1D92 A AVCBUS: .WORD   VCBUS      ; POINTER TO VEHICLE IS BUS ROUTINE
1D24 1D9B A AVCVNB: .WORD   VCVNB      ; POINTER TO VEHICLE NOT BUS ROUTINE
1D25 1DA2 A AVCER:  .WORD   VCER       ; POINTER TO OVERFLOW ERROR ROUTINE
```

```
1D26          .LIST   LVC
1D26        ;                                   G1' > G2'
1D26 860B A VCK2:   LD      AC1,D11(AC2)
1D27 0680 A VCK21:  DSUB    D8(AC2)             ; B13    FORM G1' - G2'
1D28 0008 A
1D29 C1EB A         ADD     AC0,VCC0            ; CLEAR OVERFLOW STATUS BIT
1D2A 0690 A         DIV     D1(AC2)             ; FORM (G1' - G2') / A0
1D2B 0001 A
1D2C        ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1D2C        ;                                   JUMP TO VCER
1D2C 074E A         SKSTF   D14
1D2D 2101 A         JMP     .+2
1D2E 25F6 A         JMP     @AVCER
1D2F A60D A         ST      AC1,D13(AC2)
1D30 4C00 A         LI      AC0,C0
1D31 A20C A         ST      AC0,D12(AC2)
1D32 210F A         JMP     VCK5
1D33        ;                                   G1' < G2'
1D33 8208 A VCK3:   LD      AC0,D8(AC2)         ; B13
1D34 8609 A         LD      AC1,D9(AC2)
1D35 0680 A         DSUB    D10(AC2)            ; FORM (G2' - G1') / A0
1D36 000A A
1D37 C1DD A         ADD     AC0,VCC0            ; CLEAR OVERFLOW STATUS BIT
1D38 0690 A         DIV     D1(AC2)             ; FORM (G2' - G1') / A0
1D39 0001 A
1D3A        ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1D3A        ;                                   JUMP TO VCER
1D3A 074E A         SKSTF   D14
1D3B 2101 A         JMP     .+2
1D3C 25E9 A         JMP     @AVCER
1D3D        ;                                   CONVERT QUOTIENT TO NEGATIVE AND STORE
1D3D 5101 A         CAI     AC1,C1
1D3E A60D A         ST      AC1,D13(AC2)
1D3F 4C00 A         LI      AC0,C0
1D40 5000 A         CAI     AC0,C0
1D41 A20C A         ST      AC0,D12(AC2)
1D42        ;                                   FORM C11*(T(C2) - T(C1))
1D42 8604 A VCK5:   LD      AC1,D4(AC2)         ; B0
1D43 D602 A         SUB     AC1,D2(AC2)
1D44 0780 A         MPY     D1(AC3)             ; B13
1D45 0001 A
1D46 A208 A         ST      AC0,D8(AC2)
1D47 A609 A         ST      AC1,D9(AC2)
1D48        ;                                   FORM C12 * (T(C3) - T(C2))  = F2'
1D48 8606 A         LD      AC1,D6(AC2)
1D49 D604 A         SUB     AC1,D4(AC2)
1D4A 0780 A         MPY     D2(AC3)
1D4B 0002 A
1D4C A20A A         ST      AC0,D10(AC2)
1D4D A60B A         ST      AC1,D11(AC2)
1D4E        ;                                   FORM ABS(F2' - F1')
1D4E F209 A         SKNE    AC0,D8(AC2)         ; SKIP IF UPPER F2' NOT = UPPER F1'
```

```
4460 1D4F 2103 A          JMP     .+4
4461 1D50 E208 A          SKG     AC0,D8(AC2)          ; SKIP IF UPPER F2' > UPPER F1'
4462 1D51 2125 A          JMP     VCK9
4463 1D52 210E A          JMP     VCK8
4464 1D53 65CE A          AND     AC1,VCM15
4465 1D54 F5CD A          SKNE    AC1,VCM15            ; SKIP IF BIT 15 NOT SET
4466 1D55 2108 A          JMP     VCK7
4467 1D56 85CE A          LD      AC1,VCM15
4468 1D57 6609 A          AND     AC1,D9(AC2)
4469 1D58 F5C9 A          SKNE    AC1,VCM15            ; SKIP IF BIT 15 NOT SET
4470 1D59 211D A          JMP     VCK9
4471 1D5A 860B A VCK6:    LD      AC1,D11(AC2)
4472 1D5B E609 A          SKG     AC1,D9(AC2)          ; SKIP IF LOWER F2' > LOWER F1'
4473 1D5C 211A A          JMP     VCK9
4474 1D5D 2103 A          JMP     VCK8
4475 1D5E 6609 A VCK7:    AND     AC1,D9(AC2)
4476 1D5F F5C2 A          SKNE    AC1,VCM15
4477 1D60 21F9 A          JMP     VCK6
4478 1D61        ;                                F2' > F1'
4479 1D61 4D00 A VCK8:    LI      AC1,C0               ; B0
4480 1D62 F60C A          SKNE    AC1,D12(AC2)         ; SKIP IF G2' - G1' NEGATIVE
4481 1D63 2137 A          JMP     VCVNB                ; OTHERWISE JUMP TO NON-BUS ROUTINE
4482 1D64 820A A          LD      AC0,D10(AC2)         ; B13
4483 1D65 860B A          LD      AC1,D11(AC2)
4484 1D66 06B0 A          DSUB    D8(AC2)
     1D67 0008 A
4485 1D68 C1AC A          ADD     AC0,VCC0             ; CLEAR OVERFLOW STATUS BIT
4486 1D69 0690 A          DIV     D7(AC2)
     1D6A 0007 A
4487 1D6B        ;                                CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
4488 1D6B        ;                                JUMP TO VCER
4489 1D6B 074E A          SKSTF   D14
4490 1D6C 2101 A          JMP     .+2
4491 1D6D 25B7 A          JMP     @AVCER
4492 1D6E 4C00 A          LI      AC0,C0
4493 1D6F 06A0 A          DADD    D12(AC2)
     1D70 000C A
4494 1D71 3081 A          NOP
4495 1D72 07A0 A          DADD    D5(AC3)
     1D73 0005 A
4496 1D74 2100 A          JMP     .+1
4497 1D75 1225 A          BOC     CCZP,VCVNB           ; IF RESULTS =>0, JUMP TO NON-BUS ROUTINE
4498 1D76 2114 A          JMP     VCNK
4499 1D77        ;                                F2' < F1'
4500 1D77 8208 A VCK9:    LD      AC0,D8(AC2)          ; B13
4501 1D78 8609 A          LD      AC1,D9(AC2)
4502 1D79 06B0 A          DSUB    D10(AC2)
     1D7A 000A A
4503 1D7B C199 A          ADD     AC0,VCC0             ; CLEAR OVERFLOW STATUS BIT
4504 1D7C 0690 A          DIV     D7(AC2)
     1D7D 0007 A
4505 1D7E        ;                                CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
```

```
1D7E        ;                                   JUMP TO VCER
1D7E 074E A          SKSTF   D14
1D7F 2101 A          JMP     .+2
1D80 25A4 A          JMP     @AVCER
1D81 A610 A          ST      AC1,D16(AC2)
1D82 4C00 A          LI      AC0,C0
1D83 8706 A          LD      AC1,D6(AC3)
1D84 06B0 A          DSUB    D15(AC2)
1D85 000F A
1D86 3081 A          NOP
1D87 06A0 A          DADD    D12(AC2)
1D88 000C A
1D89 2100 A          JMP     .+1
1D8A 1210 A          BOC     CCZP,VCVNB          ; IF RESULTS =>0, JUMP TO NON-BUS ROUTINE
1D8B         VCNK:
1D8B FD88 A          SKNE    AC3,VCCP            ; BO
1D8C 2101 A          JMP     .+2
1D8D 2104 A          JMP     VCBUS
1D8E 4C06 A          LI      AC0,C6
1D8F 3300 A          RADD    AC0,AC3
1D90 2500 A          JMP     @AVCK
1D91 1CF6 A AVCK:    .WORD   VCK
```

```
1D92                .LIST   LVC
1D92 4C00 A VCBUS:  LI      AC0,C0
1D93 4D19 A         LI      AC1,DPMD            ; AC0 = PREEMPT MINIMUM DURATION
1D94 B47B A         ST      AC1,@AVCBT
1D95 8C7D A         LD      AC3,AVCLOG
1D96 7809 A         ISZ     D9(AC3)
1D97 0200 A         RTS     D0
1D98 7808 A         ISZ     D8(AC3)
1D99 2100 A         JMP     .+1
1D9A 0200 A         RTS     D0
1D9B 4C01 A VCVNB:  LI      AC0,C1
1D9C 8C7D A         LD      AC3,AVCLOG
1D9D 780B A         ISZ     D11(AC3)
1D9E 0200 A         RTS     D0
1D9F 780A A         ISZ     D10(AC3)
1DA0 2100 A         JMP     .+1
1DA1 0200 A         RTS     D0
1DA2        VCER:
1DA2        ;                           ERROR DETECTED   OVERFLOW ON DIVIDE
1DA2 947E A         LD      AC1,@AVCDT
1DA3 8C77 A         LD      AC3,APERR
1DA4 2049 A         JSR     @AERE0
1DA5 4C01 A         LI      AC0,C1
1DA6 0200 A         RTS     D0
```

```
4552 1DA7                 .PAGE
4553 1DA7                 .ASECT
4554 1DA7 1E00 A          .=X'1E00
```

```
1E00                    .LIST   LFB
1E00          ;
1E00          ;                    ... BACKGROUND PROGRAMS ...
1E00          EBKD:
1E00          ;
1E00          ;                    ... RAM ADDRESS TEST ...
1E00 80D6 A   EBKD1:  LD      AC0,ADEND        ; START ADDRESS
1E01 80D4 A           ST      AC0,@ASSAD
1E02 80F6 A           LD      AC0,H3FF         ; END ADDRESS
1E03 80D5 A           ST      AC0,@ASEAD
1E04          ;
1E04 2C5A A           JSR     @ASMAD
1E05 2104 A           JMP     EBKDE            ; ERROR RETURN, (AC2) = CONTENTS
1E06                                           ;               (AC3) = ADDRESS
1E06 2100 A           JMP     .+1              ; NO ERROR RETURN
1E07          ;
1E07 0744 A           SKSTF   D4               ; SKIP IF STATUS IS SET
1E08 21F7 A           JMP     EBKD1            ; RESTART RAM ADDRESS TEST
1E09 2100 A           JMP     EBKDE            ; JUMP IF ERROR
1E0A          ;
1E0A          ;                    ... ERROR, TERMINAL WAIT LOOP ...
1E0A 2100 A   EBKDE:  JMP     .+1              ; WAIT FOR MONITOR ROUTINE TO INITIATE
1E0B 21FE A           JMP     .-1              ;    ERROR RESPONSE
```

```
1E0C                    .LIST   LEB
1E0C          ;
1E0C          ;                    ... RAM ADDRESS TEST ...
1E0C 4CFF A SMAD:   LI      AC0,CM1          ; CLEAR ERROR COUNTER
1E0D 88D5 A         LD      AC2,ASEAD        ; POINTER TO END ADDRESS
1E0E 9CD4 A         LD      AC3,@ASSAD       ; START ADDRESS
1E0F          ;
1E0F AF00 A SMAD1:  ST      AC3,(AC3)        ; STORE ADDRESS AT EACH LOCATION
1E10 4801 A         AISZ    AC3,C1           ; INCREMENT ADDRESS
1E11 EE00 A         SKG     AC3,(AC2)        ; SKIP IF DONE
1E12 21FC A         JMP     SMAD1
1E13          ;
1E13 9CD4 A         LD      AC3,@ASSAD       ; START ADDRESS
1E14 FF00 A SMAD2:  SKNE    AC3,(AC3)        ; SKIP IF MISMATCH
1E15 2106 A         JMP     SMAD3            ; MATCH, CONTENTS = ADDRESS
1E16 4801 A         AISZ    AC0,C1           ; SKIP IF FIRST ERROR
1E17 1301 A         BOC     CCB0,SMADE       ; BRANCH IF SECOND ERROR
1E18 21FB A         JMP     SMAD2            ; CONTINUE LOOP
1E19          ;
1E19          ;                    ... TWO CONSECUTIVE ADDRESS ERRORS ...
1E19 8B00 A SMADE:  LD      AC2,(AC3)        ; GET ERRONEOUS CONTENTS
1E1A 0704 A         SETST   D4               ; SET STATUS FLAG 4
1E1B 0200 A         RTS     D0               ; RETURN FROM SMAD (ERROR)
1E1C          ;
1E1C 4CFF A SMAD3:  LI      AC0,CM1          ; CLEAR ERROR COUNTER
1E1D 4F01 A         AISZ    AC3,C1           ; INCREMENT ADDRESS
1E1E EE00 A         SKG     AC3,(AC2)        ; SKIP IF DONE
1E1F 21F4 A         JMP     SMAD2            ; CONTINUE LOOP
1E20 0201 A         RTS     D1               ; RETURN FROM SMAD (NO ERROR)
```

```
4611 1E21                    .PAGE
4612 1E21                    .ASECT
4613 1E21 1F00 A             .=X'1F00
4614 1F00          ;
```

```
1F00                    .LIST   LBX
1F00          ;                    ... D A T A   I N P U T   C P C ...
1F00          ;
1F00          ;                    ... DATA INPUT FOR RTC SEQUENCE ...
1F00          ;
1F00 8CE3 A BRTC:   LD      AC3,ATCP        ; INPUT AND SAVE DISCRETE INPUT WORD
1F01 8868 A         LD      AC2,ABBUF
1F02 0403 A         RIN     FC3
1F03 A200 A         ST      AC0,D0(AC2)
1F04 8CE0 A         LD      AC3,ALDU        ; INPUT AND SAVE COMMAND REGISTERS
1F05 0403 A         RIN     FC3
1F06 A201 A         ST      AC0,D1(AC2)
1F07 0404 A         RIN     FC4
1F08 A202 A         ST      AC0,D2(AC2)
1F09          ;
1F09 8CE9 A         LD      AC3,APPDU       ; INPUT AND SAVE PROG. PANEL DISCRETES
1F0A 0403 A         RIN     FC3
1F0B B0BB A         ST      AC0,@ABPPD
1F0C 0200 A         RTS     D0
1F0D          ;
1F0D          ;
```

```
1F0D                .LIST   LBX
1F0D        ;
1F0D        ;                  ... DATA INPUT FROM PROG. PANEL ...
1F0D        ;                      (THUMBWHEEL SWITCH DATA ONLY)
1F0D        ;
1F0D        ;                  ... THIS ROUTINE IS EXECUTED DURING RESTART
1F0D        ;                         INITIALIZATION AND THEREAFTER ONLY UPON INPUT
1F0D        ;                         REQUEST VIA THE PROG PANEL PUSHBUTTON ...
1F0D        ;
1F0D 88BB A BPAN:   LD      AC2,ABPPD       ; BASE ADDRESS OF INPUT DATA BUFFER
1F0E 8CEA A         LD      AC3,APP1U       ; PLAN 1 PERIPHERAL DEVICE ADDRESS
1F0F 2909 A         JSR     BPIN            ; READ AND SAVE PARAMETER SWITCH DATA
1F10        ;
1F10 4A07 A         AISZ    AC2,C7          ; INCREMENT BY LENGTH OF BUFFER
1F11 2100 A         JMP     .+1
1F12 8CEB A         LD      AC3,APP2U       ; PLAN 2 PERIPHERAL DEVICE ADDRESS
1F13 2905 A         JSR     BPIN            ; READ AND SAVE PARAMETER SWITCH DATA
1F14        ;
1F14 4A07 A         AISZ    AC2,C7          ; INCREMENT BY LENGTH OF BUFFER
1F15 2100 A         JMP     .+1
1F16 8CEC A         LD      AC3,APP3U       ; PLAN 3 PERIPHERAL DEVICE ADDRESS
1F17 2901 A         JSR     BPIN            ; READ AND SAVE PARAMETER SWITCH DATA
1F18        ;
1F18 0200 A         RTS     D0              ; RETURN FROM BPAN
1F19        ;
1F19        ;                  ... INPUT A SET OF PROG. PANEL PARAMETER SWITCHES ...
1F19        ;                         (AC2) = BASE ADDRESS OF DATA BUFFER
1F19        ;                         (AC3) = PERIPHERAL DEVICE ADDRESS
1F19 0411 A BPIN:   RIN     FC11
1F1A A201 A         ST      AC0,D1(AC2)
1F1B 0412 A         RIN     FC12
1F1C A202 A         ST      AC0,D2(AC2)
1F1D 0413 A         RIN     FC13
1F1E A203 A         ST      AC0,D3(AC2)
1F1F 0414 A         RIN     FC14
1F20 A204 A         ST      AC0,D4(AC2)
1F21 0415 A         RIN     FC15
1F22 A205 A         ST      AC0,D5(AC2)
1F23 0410 A         RIN     FC10
1F24 A206 A         ST      AC0,D6(AC2)
1F25 0417 A         RIN     FC17
1F26 A207 A         ST      AC0,D7(AC2)
1F27        ;
1F27 0200 A         RTS     D0              ; RETURN FROM BPIN
1F28        ;
```

```
1F28                    .LIST   LBX
1F28          ;                    ... D A T A   O U T P U T   C P C ...
1F28          ;
1F28          ;                    ... DATA OUTPUT FOR RTC SEQUENCE ...
1F28 8CE0 A XRTC:   LD      AC3,ALDU
1F29 886B A         LD      AC2,AXBUF
1F2A 8201 A         LD      AC0,D1(AC2)
1F2B 0601 A         ROUT    FC1             ; OUTPUT COMMAND WORD 1
1F2C 8202 A         LD      AC0,D2(AC2)
1F2D 0602 A         ROUT    FC2             ; OUTPUT COMMAND WORD 2
1F2E 8CE4 A         LD      AC3,ASDU
1F2F 8200 A         LD      AC0,D0(AC2)
1F30 0601 A         ROUT    FC1             ; OUTPUT STATUS DISPLAY WORD
1F31 0200 A         RTS     D0
1F32          ;
1F32 8CE0 A XWDT:   LD      AC3,ALDU
1F33 4C00 A         LI      AC0,C0
1F34 0641 A         ROUT    FC41            ; RESET CONTROL PULSE FOR DWELL TIMER
1F35 0200 A         RTS     D0
1F36          ;
```

```
1F36                    .LIST   LBX
1F36          ;
1F36          ;                    ...FORMAT OUTPUT COMMAND WORDS ...
1F36 8881 A XSDC:   LD      AC2,ATBUF
1F37 8200 A         LD      AC0,D0(AC2)      ; FETCH COMMAND INDEX WORD
1F38 1512 A         BOC     CCNZ,XFOW        ; TEST FOR NON-ZERO WORD FOR NEW COMMAND
1F39 90A0 A         LD      AC0,@AECMF       ; COMMAND FEEDBACK ERROR FLAG
1F3A 1508 A         BOC     CCNZ,XCER        ; BRANCH IF ERROR
1F3B          ;                    ... (INSERT ADDITIONAL TESTS HERE)
1F3B 886B A         LD      AC2,AXBUF
1F3C 8201 A         LD      AC0,D1(AC2)
1F3D 60F0 A         AND     AC0,H7FFF        ; CLEAR UPDATE BIT IN COMMAND WORD 1
1F3E A201 A         ST      AC0,D1(AC2)
1F3F 8202 A         LD      AC0,D2(AC2)
1F40 60F0 A         AND     AC0,H7FFF        ; CLEAR UPDATE BIT IN COMMAND WORD 2
1F41 A202 A         ST      AC0,D2(AC2)
1F42 0200 A         RTS     D0
1F43          ;                    ... COMMAND ERROR, SET UPDATE BIT FOR NEXT OUTPUT ...
1F43 886B A XCER:   LD      AC2,AXBUF
1F44 8201 A         LD      AC0,D1(AC2)      ; XOCW1
1F45 60F0 A         AND     AC0,H7FFF        ; CLEAR UPDATE BIT
1F46 A201 A         ST      AC0,D1(AC2)
1F47 8202 A         LD      AC0,D2(AC2)      ; XOCW2
1F48 072F A         SETBIT  D15              ; SET UPDATE BIT
1F49 A202 A         ST      AC0,D2(AC2)
1F4A 0200 A         RTS     D0
1F4B          ;
1F4B          ;                    ... FORMAT NEW OUTPUT COMMAND WORDS ...
1F4B 6137 A XFOW:   AND     AC0,H7F00        ; EXTRACT MAIN STREET COMMAND INDEX
1F4C 5CF8 A         SHR     AC0,D8           ; SHIFT TO LOWER BITS
1F4D 1102 A         BOC     CCEZ,XMCE
1F4E E147 A         SKG     AC0,XIMX         ; TEST FOR VALID INDEX (1 TO XIMX)
1F4F 210C A         JMP     XFO1
1F50          ;
1F50          ;                    ... INVALID MAIN STREET COMMAND INDEX ...
1F50 8881 A XMCE:   LD      AC2,ATBUF
1F51 8200 A         LD      AC0,D0(AC2)      ; FETCH COMMAND INDEX WORD
1F52 886C A         LD      AC2,AXERR
1F53 A202 A         ST      AC0,D2(AC2)      ; STORE IN ERROR BUFFER, XERRO
1F54 8201 A         LD      AC0,D1(AC2)      ; FETCH ERROR COUNTER AND INCREMENT
1F55 4801 A         AISZ    AC0,C1
1F56 2100 A         JMP     .+1
1F57 A201 A         ST      AC0,D1(AC2)
1F58 8200 A         LD      AC0,D0(AC2)
1F59 0720 A         SETBIT  D0               ; SET ERROR FLAG BIT 0
1F5A A200 A         ST      AC0,D0(AC2)
1F5B 0200 A         RTS     D0
1F5C          ;
1F5C 8D27 A XFO1:   LD      AC3,AXCFT        ; FETCH ADDRESS OF COMMAND FORMAT TABLE
1F5D 3300 A         RADD    AC0,AC3          ; ADD COMMAND INDEX
1F5E 8300 A         LD      AC0,(AC3)        ; FETCH COMMAND FORMAT
1F5F 8C6B A         LD      AC3,AXBUF
```

```
1F60 A301 A          ST      AC0,D1(AC3)      ; STORE IN XOCW1
1F61          ;
1F61 8200 A          LD      AC0,D0(AC2)      ; FETCH COMMAND INDEX WORD
1F62 611F A          AND     AC0,H007F        ; EXTRACT CROSS STREET INDEX
1F63 1102 A          BOC     CCEZ,XCCE
1F64 E131 A          SKG     AC0,XIMX         ; TEST FOR VALID INDEX (1 TO XIMX)
1F65 210C A          JMP     XF02
1F66          ;
1F66          ;                  ... INVALID CROSS STREET COMMAND INDEX ...
1F66 8881 A XCCE:    LD      AC2,ATBUF
1F67 8200 A          LD      AC0,D0(AC2)      ; FETCH COMMAND INDEX WORD
1F68 886C A          LD      AC2,AXERR
1F69 A204 A          ST      AC0,D4(AC2)      ; STORE IN ERROR BUFFER, XERR1
1F6A 8203 A          LD      AC0,D3(AC2)      ; FETCH ERROR COUNTER AND INCREMENT
1F6B 4801 A          AISZ    AC0,C1
1F6C 2100 A          JMP     .+1
1F6D A203 A          ST      AC0,D3(AC2)
1F6E 8200 A          LD      AC0,D0(AC2)
1F6F 0721 A          SETBIT  D1               ; SET ERROR FLAG BIT 1
1F70 A200 A          ST      AC0,D0(AC2)
1F71 0200 A          RTS     D0
1F72          ;
1F72 8011 A XF02:    LD      AC3,AXCFT        ; FETCH ADDRESS OF COMMAND FORMAT TABLE
1F73 3300 A          RADD    AC0,AC3          ; ADD COMMAND INDEX
1F74 8300 A          LD      AC0,(AC3)        ; FETCH COMMAND FORMAT
1F75 072F A          SETBIT  D15              ; SET UPDATE BIT
1F76 8C6B A          LD      AC3,AXBUF
1F77 A302 A          ST      AC0,D2(AC3)      ; STORE IN XOCW2
1F78          ;
1F78 8881 A          LD      AC2,ATBUF
1F79 8C6B A          LD      AC3,AXBUF
1F7A 8303 A          LD      AC0,D3(AC3)      ; FETCH CURRENT AND STORE IN PREVIOUS
1F7B A304 A          ST      AC0,D4(AC3)      ;    COMMAND INDEX WORD
1F7C 8200 A          LD      AC0,D0(AC2)      ; FETCH AND SAVE CURRENT COMMAND
1F7D A303 A          ST      AC0,D3(AC3)      ;    INDEX WORD
1F7E 4C00 A          LI      AC0,C0
1F7F A200 A          ST      AC0,D0(AC2)          ; ZERO COMMAND INDEX WORD(TCIWC)
1F80 0200 A          RTS     D0
1F81          ;
1F81          ;                  ... LOCAL DATA ...
1F81 0006 A H0006:   .WORD   X'0006           ; HEX CONSTANT (BITS 1,2 SET)
1F82 007F A H007F:   .WORD   X'007F           ; HEX CONSTANT (BITS 15 - 8 RESET)
1F83 7F00 A H7F00:   .WORD   X'7F00           ; HEX CONSTANT (BITS 15 AND 7 - 0 RESET)
1F84          ;
```

```
4807 1F84                .LIST   LBX
4808 1F84        ;
4809 1F84        ;                    ... COMMAND FORMAT TABLE ...
4810 1F84 1F84 A AXCFT:  .WORD   XCFT-1
4811 1F85        XCFT:
4812 1F85        ;                                          L
4813 1F85        ;                                          T          S
4814 1F85        ;                                                     E
4815 1F85        ;                                          P   S      C   P
4816 1F85        ;                                      U   R   P      O   R
4817 1F85        ;                                      P   I   E      N   I
4818 1F85        ;                                      D   M   C      D   M
4819 1F85        ;                                      A   A   I  P   A   A
4820 1F85        ;                                      T   R   A  E   R   R
4821 1F85        ;                                      E   Y   L  D   Y   Y
4822 1F85        ;
4823 1F85        ;                                              C   C   C   C
4824 1F85        ;                                              LS  LS  LS  LS
4825 1F85        ;                                              ET  ET  ET  ET
4826 1F85        ;                                COMMAND    GG GAO GAO GAO GAO
4827 1F85        ;                                 INDEX     OO ORP ORP ORP ORP
4828 1F85        ;
4829 1F85        ;                                      BITS SET = 1 (CIRCUITS ON)
4830 1F85        ;                                      1 111 11
4831 1F85        ;                                      5 432 109 876 543 210
4832 1F85        ;
4833 1F85        ;      ... PBD33J ...
4834 1F85 0249 A          .WORD   X'0249          ; 1          S   S   S   S
4835 1F86 0324 A          .WORD   X'0324          ; 2          S G   G   G     G GO
4836 1F87 02A4 A          .WORD   X'02A4          ; 3          S  C  G   G     C CLEAR
4837 1F88 0264 A          .WORD   X'0264          ; 4          S   S G   G     S STOP
4838 1F89 0252 A          .WORD   X'0252          ; 5          S   S  C   C
4839 1F8A        ;
4840 1F8A        ;      ... PBD14C ...
4841 1F8A 1124 A          .WORD   X'1124          ; 6       G     G   G   G
4842 1F8B 10A4 A          .WORD   X'10A4          ; 7       G      C  G   G
4843 1F8C 1094 A          .WORD   X'1094          ; 8       G      C   C  G
4844 1F8D 304C A          .WORD   X'304C          ; 9      GG       S   S G
4845 1F8E 044A A          .WORD   X'044A          ; A         C    S   S  C
4846 1F8F 0921 A          .WORD   X'0921          ; B          G   G   G     S
4847 1F90 1121 A          .WORD   X'1121          ; C       G     G   G     S
4848 1F91 0249 A          .WORD   X'0249          ; D          S   S   S   S
4849 1F92 0041 A          .WORD   X'0041          ; E              S       S
4850 1F93 0104 A          .WORD   X'0104          ; F             G       G
4851 1F94 0084 A          .WORD   X'0084          ; 10             C      G
4852 1F95 0082 A          .WORD   X'0082          ; 11             C       C
4853 1F96        ;
4854 1F96        XCFTE:                           ; END OF COMMAND FORMAT TABLE
4855 1F96 0011 A XIMX:   .WORD   XCFTE-XCFT       ; MAXIMUM VALUE OF COMMAND INDEX
4856 1F97        ;
```

```
1F97                    .LIST   LBX
1F97            ;
1F97            ;                  ... TURN STATUS LIGHTS ON/OFF ...
1F97            ;                         UPON ENTRY, AC3 CONTAINS PERIPHERAL
1F97            ;                                     DEVICE ADDRESS
1F97            ;
1F97            ;                  ... TURN STATUS LIGHT 1 ON ...
1F97            XLON1:
1F97 4C00 A             LI      AC0,C0
1F98 0644 A             ROUT    FC44            ; OUTPUT CONTROL PULSE
1F99 0200 A             RTS     D0              ; RETURN FROM XLON1
1F9A            ;
1F9A            ;                  ... TURN STATUS LIGHT 1 OFF ...
1F9A            XLOF1:
1F9A 4C00 A             LI      AC0,C0
1F9B 0645 A             ROUT    FC45            ; OUTPUT CONTROL PULSE
1F9C 0200 A             RTS     D0              ; RETURN FROM XLOF1
1F9D            ;
1F9D            ;                  ... TURN STATUS LIGHT 2 ON ...
1F9D            XLON2:
1F9D 4C00 A             LI      AC0,C0
1F9E 0646 A             ROUT    FC46            ; OUTPUT CONTROL PULSE
1F9F 0200 A             RTS     D0              ; RETURN FROM XLON2
1FA0            ;
1FA0            ;                  ... TURN STATUS LIGHT 2 OFF ...
1FA0            XLOF2:
1FA0 4C00 A             LI      AC0,C0
1FA1 0647 A             ROUT    FC47            ; OUTPUT CONTROL PULSE
1FA2 0200 A             RTS     D0              ; RETURN FROM XLOF2
```

```
4888 1FA3                    .PAGE
4889 1FA3                    .ASECT
4890 1FA3 3000 A             .=X'3000
4891 3000          ;
```

```
4894 3000                .LIST   LDB
4895 3000        ;
4896 3000        ;                   ... D A T A   B A S E ...
4897 3000        ;
4898 3000        ;
4899 3000        ;                   ... BUS DETECTOR SIGNATURE DATA ...
4900 3000        ;
4901 3000        ;        ( UP TO 32 DETECTORS @ 8 WORDS PER SIGNATURE)
4902 3000        ;
4903 3000        ;                       WORD 0 - TOTAL SIGNATURE DURATION
4904 3000        ;                            1 - AMPLITUDE OF FIRST PEAK
4905 3000        ;                            2 - TIME OF FIRST PEAK
4906 3000        ;                            3 - AMPLITUDE OF SECOND PEAK
4907 3000        ;                            4 - TIME OF SECOND PEAK
4908 3000        ;                            5 - AMPLITUDE OF THIRD OR LAST PEAK
4909 3000        ;                            6 - TIME OF LAST PEAK
4910 3000        ;                            7 - END OF SIGNATURE STATUS
4911 3000        ;
4912 3000        VCSDB:
4913 3000 3100 A         .=.+256
4914 3100        ;
4915 3100        ;                   ... DATA INPUT CPC DATA BASE ...
4916 3100        BBUF:
4917 3100 3101 A BDIN:   .=.+1                   ; DISCRETE INPUT DATA WORD
4918 3101 3102 A BCR1:   .=.+1                   ; COMMAND REGISTER 1 STATUS
4919 3102 3103 A BCR2:   .=.+1                   ; COMMAND REGISTER 2 STATUS
4920 3103        ;
4921 3103        ;                   ... PROG PANEL INPUT BUFFER ...
4922 3103        ;
4923 3103        ;                   ... DISCRETE SWITCHES ...
4924 3103        ;
4925 3103        ;                               DIAL 1
4926 3103        ;                          BIT 0, BUS PREEMPT
4927 3103        ;                          BIT 1, MASTER CONTROLLER SYNCH
4928 3103        ;
4929 3103        ;                               DIAL 2
4930 3103        ;                          BIT 4, BUS PREEMPT
4931 3103        ;                          BIT 5, MASTER CONTROLLER SYNCH
4932 3103        ;
4933 3103        ;                               DIAL 3
4934 3103        ;                          BIT 8, BUS PREEMPT
4935 3103        ;                          BIT 9, MASTER CONTROLLER SYNCH
4936 3103        ;
4937 3103        ;                          BIT 10, INPUT REQUEST
4938 3103        ;                          BIT 11, MANUAL OVERRIDE
4939 3103 3104 A BPPD:   .=.+1
4940 3104        ;
4941 3104        ;                   ... THUMBWHEEL INPUT DATA ...
4942 3104        ;
4943 3104        ;                          D0, CYCLE LENGTH
4944 3104        ;                          D1, OFFSET 1
4945 3104        ;                          D2, OFFSET 2
```

```
3104          ;                          D3, OFFSET 3
3104          ;                          D4, MSG SPLIT (%)
3104          ;                          D5, INT 1
3104          ;                          D6, INT 2
3104 310B A BPD1:   .=.+7                      ; DIAL 1
310B 3112 A BPD2:   .=.+7                      ; DIAL 2
3112 3119 A BPD3:   .=.+7                      ; DIAL 3
3119          ;
3119          ;                     ... DATA OUTPUT CPC DATA BASE ...
3119          XBUF:
3119 311A A XSDW:   .=.+1                      ; STATUS DISPLAY WORD
311A 311B A XOCW1:  .=.+1                      ; OUTPUT COMMAND WORD 1 (MAIN STREET)
311B 311C A XOCW2:  .=.+1                      ; OUTPUT COMMAND WORD 2 (CROSS STREET)
311C 311D A XCIWC:  .=.+1                      ; CURRENT COMMAND INDEX WORD
311D 311E A XCIWP:  .=.+1                      ; PREVIOUS COMMAND INDEX WORD
311E          ;
311E          ;                     ... PREEMPT CPC DATA BASE ...
311E          PBUF:
311E 311F A PIP:    .=.+1                          ; PREEMPT IN PROGRESS FLAG
311F 3120 A PTEM:   .=.+1                          ; TEMPORARY STORAGE LOCATION
3120 3121 A PTCP:   .=.+1                      ; PRIMARY CONTRACTIBLE INT ADDRESS
3121 3122 A PTCS:   .=.+1                      ; SECONDARY CONTRACTIBLE INT ADDRESS
3122 3123 A PTEI:   .=.+1                      ; EXTENDIBLE INT ADDRESS
3123 3126 A PMIF:   .=.+3                          ; PREEMPT MONITOR INDEX FLAGS
3126 3127 A PFREQ:  .=.+1                          ; PREEMPT REQUEST FLAG
3127 3128 A PMDT:   .=.+1                          ; PREEMPT MINIMUM DURATION TIMER
3128 3129 A PTAVT:  .=.+1                          ; AVAILIBLE TIME FOR INTERVAL RESTORATION
3129 312A A PTITE:  .=.+1                          ; INTERVAL TO BE RESTORED
312A 312B A PTPMT:  .=.+1                          ; FIRST LOCATION OF MASTER TIMING PLAN
312B          ;
312B          ;                     ... MAXIMUM PREEMPT EXTENSION PARAMETERS ...
312B 312C A PEMAX:  .=.+1                      ; CURRENT DIAL
312C 312D A         .=.+1                      ; DIAL 1
312D 312E A         .=.+1                      ; DIAL 2
312E 312F A         .=.+1                      ; DIAL 3
312F          ;
312F          ;                     ... VEHICLE CLASSIFIER DATA ...
312F 3130 A VCDT:   .=.+1                          ; VEHICLE DETECTOR INDEX
3130 3131 A VCBT:   .=.+1                      ; TIMER FOR RECENT BUS CLASS.
3131          ;
3131 3151 A VCVC:   .=.+32                     ; VEHICLE COUNTERS (ONE PER DETECTOR)
3151 3165 A VCDB:   .=.+20
3165          ;
3165          ;                     ... VEHICLE CLASSIFIER VEHICLE LOG ...
3165          ;
3165          ;                          DOUBLE PRECISION, UNSIGNED WORDS
3165          ;
3165          VCLOG:
3165 3167 A         .=.+2                          ; NUMBER OF SIGNATURES REJECTED AS NOISE
3167 3169 A         .=.+2                          ; NUMBER OF GROSS TEST FAILURES
3169 316B A         .=.+2                          ; NUMBER OF VEHICLES HANDLED BY
316B          ;                                      STOPPED VEHICLE CLASSIFIER
```

```
316B 316D A          .=.+2                    ; NUMBER OF VEHICLES HANDLED BY
316D          ;                                 MOVING VEHICLE CLASSIFIER
316D 316F A          .=.+2                    ; NUMBER OF VEHICLES CLASSIFIED
316F          ;                                 AS BUSES
316F 3171 A          .=.+2                    ; NUMBER OF VEHICLES CLASSIFIED
3171          ;                                 AS NON-BUSES
3171          ;
3171          ;                    ... TIMING PLAN CPC DATA BASE ...
3171          TBUF:
3171 3172 A TCIWC:   .=.+1                    ; COMMAND INDEX WORD BUFFER
3172 3173 A TCDS:    .=.+1                    ; CURRENT DIAL SELECTION
3173 3174 A TIN:     .=.+1                    ; CURRENT INTERVAL INDEX
3174 3175 A TINT:    .=.+1                    ; CURRENT INTERVAL TIMER
3175 3176 A TFST:    .=.+1                    ; WAIT FOR SYNC START FLAG
3176 3177 A TEOC:    .=.+1                    ; END OF CYCLE FLAG
3177 3178 A TEMP:    .=.+1                    ; TEMPORARY LOCATION
3178 3179 A TDWL:    .=.+1                    ; DIAL 1, MAX DWELL FOR SYNC START
3179 317A A          .=.+1                    ; DIAL 2, MAX DWELL FOR SYNC START
317A 317B A          .=.+1                    ; DIAL 3, MAX DWELL FOR SYNC START
317B 318F A TPM1:    .=.+20                   ; MASTER TP TABLE, DIAL 1
318F 31A3 A TPM2:    .=.+20                   ; MASTER TP TABLE, DIAL 2
31A3 31B7 A TPM3:    .=.+20                   ; MASTER TP TABLE, DIAL 3
31B7 31CB A TNTP:    .=.+20                   ; NEXT TP TABLE
31CB 31DF A TCTP:    .=.+20                   ; CURRENT TP TABLE
31DF          ;
31DF          ;                    ... EXECUTIVE CPC DATA BASE ...
31DF          ;
31DF          EBUF:
31DF 31E0 A ERTCF:   .=.+1                    ; RTC COMPUTE FLAG
31E0          ;
31E0 31E1 A ETST:    .=.+1                    ; 'WAIT FOR SYNCH START' DWELL TIMER
31E1          ;
31E1 31E2 A ETCY:    .=.+1                    ; CURRENT CYCLE TIMER
31E2 31E3 A          .=.+1                    ; PREVIOUS MAXIMUM
31E3 31E4 A          .=.+1                    ; PREVIOUS MINIMUM
31E4 31E5 A EPST:    .=.+1                    ; START PULSE INDICATOR
31E5 31E6 A EPDT:    .=.+1                    ; START PULSE DURATION TIMER
31E6          ;
31E6 31E7 A ECMF:    .=.+1                    ; TEMPORARY FLAG FOR ECOM
31E7          ESYD:                           ; SYNCH DATA BUFFER (DIAL 1,2,3)
31E7                                          ;    WORD 0 - CYCLE LENGTH
31E7                                          ;    WORD 1 - RESET 1 POSITION ON DIAL
31E7                                          ;    WORD 2 - RESET 2 POSITION ON DIAL
31E7                                          ;    WORD 3 - RESET 3 POSITION ON DIAL
31E7                                          ;    WORD 4 - END OF START PULSE (3 %)
31E7                                          ;    WORD 5 - SYNCH DIAL TIMER
31E7 31ED A          .=.+6                    ; DIAL 1
31ED 31F3 A          .=.+6                    ; DIAL 2
31F3 31F9 A          .=.+6                    ; DIAL 3
31F9          ;
31F9          ;                    ... LOOP MASTER INTERCONNECT DATA ...
31F9 31FA A ER123:   .=.+1                    ; RESET SELECT
```

```
5050 31FA 31FB A ERNO:    .=.+1          ; ERNO, OTH PAST VALUE OF R1,R2,R3
5051 31FB 31FC A          .=.+1          ; ERN1, 1ST PAST VALUE OF R1,R2,R3
5052 31FC 31FD A          .=.+1          ; ERN2, 2ND PAST VALUE OF R1,R2,R3
5053 31FD         ;
5054 31FD 31FE A ED23:    .=.+1          ; DIAL SELECT
5055 31FE 31FF A EDNO:    .=.+1          ; EDNO, OTH PAST VALUE OF D2,D3
5056 31FF 3200 A          .=.+1          ; EDN1, 1ST PAST VALUE OF D2,D3
5057 3200 3201 A          .=.+1          ; EDN2, 2ND PAST VALUE OF D2,D3
5058 3201         ;
5059 3201 3202 A EDMIT:   .=.+1          ; DETECTOR MONITORING INTERVAL TIMER
5060 3202 3203 A EDETT:   .=.+1          ; TEMP STORAGE FOR DET MONITORING
5061 3203         ;
```

```
5064 3203                .LIST   LDB
5065 3203        ;
5066 3203        ;                    ... SAVE AREA FOR RTC INTERRUPTS ...
5067 3203        ESAV:
5068 3203 3204 A         .=.+1                    ; AC0
5069 3204 3205 A         .=.+1                    ; AC1
5070 3205 3206 A ESV2:   .=.+1                    ; AC2
5071 3206 3207 A         .=.+1                    ; AC3
5072 3207 3208 A ESVF:   .=.+1                    ; RALU FLAGS
5073 3208 3218 A         .=.+16                   ; STACK SAVE AREA
5074 3218 3219 A ESVS:   .=.+1                    ; SELECT FLAG STATE
5075 3219        ;
5076 3219        ;                    ... SNAPSHOT (COTF) SAVE AREA ...
5077 3219        CSAV:
5078 3219 321A A         .=.+1                            ; AC0
5079 321A 321B A         .=.+1                            ; AC1
5080 321B 321C A CSV2:   .=.+1                            ; AC2
5081 321C 321D A         .=.+1                            ; AC3
5082 321D 321E A CSVF:   .=.+1                            ; RALU FLAGS
5083 321E 322E A         .=.+16                           ; STACK SAVE AREA
5084 322E 322F A CSVS:   .=.+1                            ; SELECT FLAG STATE
5085 322F        ;
5086 322F        ;                    ... STACKFULL SAVE AREA ...
5087 322F        ISV:
5088 322F 3230 A ISV0:   .=.+1
5089 3230 3231 A ISV1:   .=.+1
5090 3231 3232 A ISV2:   .=.+1
5091 3232 3233 A ISV3:   .=.+1
5092 3233 3234 A ISTK1:  .=.+1
5093 3234 3235 A ISTK2:  .=.+1
5094 3235 3236 A ISTK3:  .=.+1
5095 3236 3237 A ISTK4:  .=.+1
5096 3237 3243 A ISTK5:  .=.+12
5097 3243        ;
5098 3243        ;                    ... DISCONNECT SAVE AREA ...
5099 3243        EDSAV:
5100 3243 3244 A         .=.+1                    ; AC0
5101 3244 3245 A         .=.+1                    ; AC1
5102 3245 3246 A EDSV2:  .=.+1                    ; AC2
5103 3246 3247 A         .=.+1                    ; AC3
5104 3247 3248 A EDSVF:  .=.+1                    ; RALU FLAGS
5105 3248 3258 A         .=.+16                   ; STACK SAVE AREA
5106 3258 3259 A EDSVS:  .=.+1                    ; SELECT FLAG STATE
5107 3259        ;
5108 3259        ;                    ... OPTIONAL USE SAVE AREA ...
5109 3259 3261 A CTST:   .=.+8                            ; OPTIONAL USE SAVE AREA
```

```
3261                .LIST   LDB
3261        ;
3261        ;                   ... PROG. PANEL DATA ...
3261        ;
3261        ;                                PANEL ROUTINE WORKING BUFFER
3261 3268 A EPD1:   .=.+7                         ; (SAME ALLOCATION AS BPD1)
3268 326F A EPD2:   .=.+7                         ; (SAME ALLOCATION AS BPD2)
326F 3276 A EPD3:   .=.+7                         ; (SAME ALLOCATION AS BPD3)
3276        ;
3276        ;                                PANEL CONVERTED DATA UPDATE BUFFER
3276 3277 A EPCD1:  .=.+1                         ; FIRST ADJ. INT. LENGTH, DIAL 1
3277 3278 A         .=.+1                         ; SECOND ADJ. INT. LENGTH, DIAL 1
3278 3279 A         .=.+1                         ; DELTA EMAX,              DIAL 1
3279 327A A         .=.+1                         ; CYCLE LENGTH             DIAL 1
327A 327B A         .=.+1                         ; RESET 1                  DIAL 1
327B 327C A         .=.+1                         ; RESET 2                  DIAL 1
327C 327D A         .=.+1                         ; RESET 3                  DIAL 1
327D 327E A         .=.+1                         ; START PULSE DURATION     DIAL 1
327E 327F A         .=.+1                         ; MAX DWELL AT MSG         DIAL 1
327F 3280 A         .=.+1                         ; INT2                     DIAL 1
3280 328A A EPCD2:  .=.+10                        ;                          DIAL 2
328A 3294 A EPCD3:  .=.+10                        ;                          DIAL 3
3294        ;
3294        ;                                FIXED TIME IN CYCLE
3294 3295 A EPFT:   .=.+1                         ; SUM OF FIXED INTERVALS   DIAL 1
3295 3296 A         .=.+1                         ; SUM OF FIXED INTERVALS   DIAL 2
3296 3297 A         .=.+1                         ; SUM OF FIXED INTERVALS   DIAL 3
3297        ;
3297 3298 A EP1E:   .=.+1                       ; ERROR FLAG FOR PLAN 1
3298 3299 A EP2E:   .=.+1                       ; ERROR FLAG FOR PLAN 2
3299 329A A EP3E:   .=.+1                       ; ERROR FLAG FOR PLAN 3
329A 329B A EPDVE:  .=.+1                       ; ERROR FLAG FOR DATA VALIDITY
329B        ;
329B 329C A EPUF:   .=.+1                       ; PROG PANEL UPDATE FLAG
329C 329D A EPCF:   .=.+1                       ; PROG PANEL COMPUTATION FLAG
329D 329E A EPIR:   .=.+1                       ; STATE OF PROG PANEL INPUT REQ SWITCH
```

```
5150 329E                .LIST   LDB
5151 329E        ;
5152 329E        ;                   ... SELF TEST DATA BASE ...
5153 329E 329F A SSAD:   .=.+1                    ; RAM ADDRESS TEST START ADDRESS
5154 329F 32A0 A SEAD:   .=.+1                    ; RAM ADDRESS TEST END ADDRESS
5155 32A0 32BE A         .=.+30                   ; (EXPANSIBILITY)
5156 32BE        ;
```

```
32DE                    .LIST   LDB
32DE            ;
32DE            ;                       ... E R R O R   B U F F E R ...
32DE            ;
32DE            ;                       ... DATA INPUT CPC ERROR FLAGS, COUNTERS, DATA ...
32DE            BEPR:
32DE 32BF A     BERRF:  .=.+1                   ; ERROR FLAGS
32BF            ;
32BF            ;                       ... DATA OUTPUT CPC ERROR FLAGS, COUNTERS, DATA ...
32BF            XEPR:
32BF 32C0 A     XERRF:  .=.+1                   ; ERROR FLAGS (SET CONDITION, BIT = 1)
32C0            ;                                   BIT 0, INVALID COMMAND INDEX FOR
32C0            ;                                          MAIN STREET
32C0            ;                                   BIT 1, INVALID COMMAND INDEX FOR
32C0            ;                                          CROSS STREET
32C0 32C2 A     XERR0:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32C2 32C4 A     XERR1:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32C4            ;
32C4            ;                       ... PREEMPT CPC ERROR FLAGS, COUNTERS, DATA ...
32C4            PERR:
32C4 32C5 A     PERRF:  .=.+1                   ; ERROR BUFFER FLAG (SET CONDITION, BIT = 1)
32C5            ;                                   BIT 0, ARITHMETIC OVERFLOW
32C5            ;                                   BIT 1, OUT OF RANGE (VCLASS DETECTOR INDEX)
32C5            ;                                   BIT 2, DATA EXCEEDS BOUNDS (VEHICLE AMP)
32C5            ;                                   BIT 3, DATA EXCEEDS BOUNDS (VEHICLE TIME)
32C5            ;                                   BIT 4, DATA SEQUENCE ERROR (VEHICLE TIME)
32C5            ;                                   BIT 5, DATA VALIDITY ERROR (NOISE)
32C5            ;                                   BIT 6, EOS STATUS WORD ALTERED DURING COPY
32C5            ;                                          OPERATION
32C5            ;                                   BIT 7, SIGNATURE LOST DURING COPY OPERATION
32C5            ;                                   BIT 8, OUT OF RANGE (PREQ DETECTOR INDEX)
32C5            ;                                ... PBD33J ...
32C5            ;                                   BIT 9, DATA VALIDITY ERROR ( ATTEMPT
32C5            ;                                          TO PREEMPT DOWNSTREAM VEHICLE)
32C5            ;                                ... PBD14C ...
32C5            ;                                   BIT 9, NON-POSITIVE DURATION OF SOURCE
32C5            ;                                          EXCHANGE INTERVAL AT PREEMPT MOD
32C5            ;
32C5 32D9 A             .=.+20                      ; (ERROR COUNTER, ERROR WORD)
32D9            ;
32D9            ;                       ... TIMING PLAN CPC ERROR FLAGS, COUNTERS, DATA ...
32D9            TERR:
32D9 32DA A     TERRF:  .=.+1                   ; ERROR FLAGS (SET CONDITION, BIT = 1)
32DA            ;                                   BIT 0, DIAL SELECT ERROR
32DA            ;                                   BIT 1, INTERVAL INDEX OUT OF RANGE
32DA 32DC A     TERR0:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32DC 32DE A     TERR1:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32DE            ;
32DE            ;                       ... EXECUTIVE CPC ERROR FLAGS, COUNTERS, DATA ...
32DE            EERR:
32DE 32DF A     EERRF:  .=.+1                   ; ERROR FLAGS (SET CONDITION, BIT = 1)
32DF            ;                                   BIT 0, RTC ERROR
```

```
5211 32DF          ;                                   BIT 1, DCR1.NE.DOCW1
5212 32DF          ;                                   BIT 2, DCR2.NE.DOCW2
5213 32DF          ;                                   BIT 3, MORE THAN ONE RESET HIGH
5214 32DF          ;                                   BIT 4, DETECTOR UNDERCOUNTING
5215 32DF                                        ;            (TOTAL FOR ALL DETECTORS,
5216 32DF                                        ;             INDEX OF LAST DETECTOR)
5217 32DF          ;                                   BIT 5, INVALID DET INDEX
5218 32DF 32E1 A EERR0:  .=.+2                   ; ERROR COUNTER, ERROR WORD
5219 32E1 32E3 A EERR1:  .=.+2                   ; ERROR COUNTER, ERROR WORD
5220 32E3 32E5 A EERR2:  .=.+2                   ; ERROR COUNTER, ERROR WORD
5221 32E5 32E7 A EERR3:  .=.+2                   ; ERROR COUNTER, ERROR WORD
5222 32E7 32E9 A EERR4:  .=.+2                   ; ERROR COUNTER, ERROR WORD
5223 32E9 32EB A EERR5:  .=.+2                   ; ERROR COUNTER, ERROR WORD
5224 32EB          ;
5225 32EB          ;                        ... PROG. PANEL ERROR DATA ...
5226 32EB          ;
5227 32EB          ;                            ERROR FLAGS (SET CONDITION, BIT = 1)
5228 32EB          ;                               BIT 0, ERROR IN ADJ INT OF TP MASTER
5229 32EB          ;                               BIT 1, DATA VALIDITY ERROR (CYCLE LENGTH)
5230 32EB          ;                               BIT 2, DATA VALIDITY ERROR (OFFSET 1)
5231 32EB          ;                               BIT 3, DATA VALIDITY ERROR (OFFSET 2)
5232 32EB          ;                               BIT 4, DATA VALIDITY ERROR (OFFSET 3)
5233 32EB          ;                               BIT 5, DATA VALIDITY ERROR (MSG SPLIT %)
5234 32EB          ;                               BIT 6, DATA VALIDITY ERROR (INT1)
5235 32EB          ;                               BIT 7, DATA VALIDITY ERROR (INT2)
5236 32EB          ;                               BIT 8,  OUT OF RANGE (CYCLE LENGTH)
5237 32EB          ;                               BIT 9,  OUT OF RANGE (OFFSET 1)
5238 32EB          ;                               BIT 10, OUT OF RANGE (OFFSET 2)
5239 32EB          ;                               BIT 11, OUT OF RANGE (OFFSET 3)
5240 32EB          ;                               BIT 12, OUT OF RANGE (INT1)
5241 32EB          ;
5242 32EB          ;
5243 32EB 32EC A EPEF1:  .=.+1                   ; ERROR FLAGS
5244 32EC 3306 A         .=.+26                  ; (ERROR COUNTER, ERROR WORD)
5245 3306          ;
5246 3306 3307 A EPEF2:  .=.+1                   ; ERROR FLAGS
5247 3307 3321 A         .=.+26                  ; (ERROR COUNTER, ERROR WORD)
5248 3321          ;
5249 3321 3322 A EPEF3:  .=.+1                   ; ERROR FLAGS
5250 3322 333C A         .=.+26                  ; (ERROR COUNTER, ERROR WORD)
5251 333C          ;
5252 333C          ;                        ... DETECTOR ERROR COUNTERS, DATA ...
5253 333C          EDERR:
5254 333C 337C A         .=.+64                  ; (ERROR COUNTER, ERROR DATUM)
5255 337C                                        ;    FOR 32 DETECTORS
5256 337C          ;
5257 337C          ;
5258 337C 337D A DEND:   .=.+1                   ; LAST LOCATION IN READ/WRITE DATA BASE
```

```
5259 337D                    .PAGE   '  '
5260 337D                    .LIST   1
5261 337D 0001 A LAC         =       1
5262 337D 0001 A LBP         =       1
5263 337D 0001 A LER         =       1
5264 337D 0001 A LES         =       1
5265 337D 0001 A LPP         =       1
5266 337D 0001 A LTP         =       1
5267 337D 0001 A LPR         =       1
5268 337D 0001 A LVC         =       1
5269 337D 0001 A LEB         =       1
5270 337D 0001 A LBX         =       1
5271 337D 0001 A LDB         =       1
5272 337D 0001 A LDU         =       1
5273 337D                    .EXTD
5274 337D                    .END

  ******      0 ERRORS IN ASSEMBLY   ******
```

```
ABBUF  ABDIN  ABERR  ABLSMU ABPAN  ABPD1  ABPD2  ABPD3  ABPPD  ABRTC
0068 A 0069 A 006A A 00E1 A 003E A 00BC A 00BD A 00BE A 00BB A 003C A

AC0    AC1    AC2    AC3    ACCU   ACCUI  ACDISP ACMU   ACOTF  ACRAM
0000 A 0001 A 0002 A 0003 A 00E8 A 10DA A 10DC A 00E2 A 003D A 10DD A

ACSAV  ACSV2  ACSVF  ACSVS  ACTST  ACWAP  ADEND  ADMU   AE10   AE100
00AE A 00AF A 00B0 A 00B1 A 00BA A 10DE A 00D6 A 00E5 A 00CC A 00CB A

AEADC  AEBKD  AEBUF  AECCU  AECCUR AECER  AECMF  AECOM  AED23  AEDERR
002F A 0024 A 009A A 0022 A 10DF A 0025 A 00A0 A 0028 A 00A4 A 00A8 A

AEDET  AEDETT AEDIS  AEDMIT AEDNO  AEDSV2 AEDSVF AEDSVS AEFOC  AEERR
002E A 00A7 A 0027 A 00A6 A 00A5 A 00B7 A 00B8 A 00B9 A 002C A 009F A

AECIS  AEM10  AEM24  AEMAD  AEMIC  AEP1E  AEP2E  AEP3E  AEPAN  AEPBI2
0021 A 0047 A 0048 A 0023 A 002A A 00CD A 00CE A 00CF A 003F A 0045 A

AEPBI3 AEPCD1 AEPCD2 AEPCD3 AEPCF  AEPCY  AEPD1  AEPD2  AEPD3  AEPDT
0046 A 00C2 A 00C3 A 00C4 A 00C7 A 00C9 A 00BF A 00C0 A 00C1 A 009E A

AEPDVE AEPEF1 AEPEF2 AEPEF3 AEPFT  AEPIN1 AEPIR  AEPP1  AEPP2  AEPP3
00D0 A 00D1 A 00D2 A 00D3 A 00C5 A 00CA A 00C8 A 0041 A 0042 A 0043 A

AEPPM  AEPST  AEPUF  AEPUP  AER123 AERAM  AERE0  AERE1  AERF10 AERE11
0040 A 009D A 00C6 A 0044 A 00A2 A 00A9 A 0049 A 004A A 0053 A 0054 A

AERE12 AERE2  AERE3  AERE4  AERE5  AERE6  AERE7  AERE8  AERF9  AERNO
0055 A 004B A 004C A 004D A 004E A 004F A 0050 A 0051 A 0052 A 00A3 A

AERSI  AESAV  AESCH  AESDW  AESV2  AESVF  AESVS  AESYD  AESYN  AETCY
0020 A 00AA A 0026 A 002D A 00AB A 00AC A 00AD A 00A1 A 0029 A 009B A

AETIM  AETST  AISTK1 AISTK2 AISTK3 AISTK4 AISV2  ALDU   APBUF  APCIM
002B A 009C A 00B3 A 00B4 A 00B5 A 00B6 A 00B2 A 00E0 A 006D A 0034 A

APEMAX APERR  APFREQ APIP   APMDT  APMIF  APMON  APP1U  APP2U  APP3U
0076 A 0077 A 0074 A 006E A 0075 A 0073 A 0030 A 00EA A 00FB A 00EC A

APPDU  APREQ  APTAVT APTCP  APTCS  APTEI  APTEM  APTER  APTITE APTPMT
00E9 A 0032 A 0078 A 0070 A 0071 A 0072 A 006F A 0033 A 0079 A 007A A

ARTC   ASDB   ASDU   ASEAD  ASMAD  ASSAD  ATAI1  ATAI2  ATBUF  ATCDS
00E7 A 00E6 A 00E4 A 00D5 A 005A A 00D4 A 0093 A 0094 A 0081 A 0083 A

ATCIW  ATCIWC ATCOPY ATCP   ATCTP  ATDIAL ATDWL  ATECI  ATEMP  ATEOC
0097 A 0082 A 0036 A 00E3 A 008F A 0038 A 0089 A 0096 A 0088 A 0087 A

ATERR  ATFST  ATHI   ATIMX  ATIN   ATINT  ATNCY  ATNTP  ATPF   ATPEX
0090 A 0086 A 0092 A 0095 A 0084 A 0085 A 0037 A 008E A 0099 A 0035 A
```

```
ATPM1  ATPM2  ATPM3  ATPP   ATPTAB ATSCI  AVCBT  AVCBUS AVCDB  AVCDT
008B A 008C A 008D A 008A A 0098 A 0091 A 007B A 1D23 A 007F A 007E A

AVCEP  AVCK   AVCLAS AVCLOG AVCSDB AVCVC  AVCVNB AXBUF  AXCFT  AXERR
1D25 A 1D91 A 0031 A 007D A 0080 A 007C A 1D24 A 006B A 1F84 A 006C A

AXLOF1 AXLOF2 AXLON1 AXLON2 AXRTC  AXSDC  AXWDT  BBUF   BCR1   BCR2
0057 A 0059 A 0056 A 0058 A 0039 A 003B A 003A A 3100 A 3101 A 3102 A

BDIN   BERR   BERRF  BPAN   BPD1   BPD2   BPD3   BPIN   BPPD   BRTC
3100 A 32BE A 32BE A 1F0D A 3104 A 310B A 3112 A 1F19 A 3103 A 1F00 A

C0     C1     C125   C15    C2     C3     C4     C48    C5     C6
0000 A 0001 A 007D A 000F A 0002 A 0003 A 0004 A 0030 A 0005 A 0006 A

C7     C8     CCB0   CCB1   CCCF   CCCU   CCEZ   CCNZ   CCPH   CCSF
0007 A 0008 A 0003 A 0004 A 000E A 000C A 0001 A 0005 A 000D A 0008 A

CCZN   CCZP   CM1    CM10   CM100  CM110  CM16   CM2    CM20   CM6
000B A 0002 A FFFF A FFF6 A FF9C A FF92 A FFF0 A FFFE A FFEC A FFFA A

COTF   COTF1  COTF2  COTF3  COTF4  COTF5  CSAV   CSV2   CSVF   CSVS
1143 A 114E A 115F A 1161 A 1172 A 1175 A 3219 A 321B A 321D A 322E A

CTST   D0     D1     D10    D11    D116   D118   D12    D13    D14
3259 A 0000 A 0001 A 000A A 000B A 0074 A 0076 A 000C A 000D A 000E A

D15    D16    D17    D18    D19    D2     D20    D21    D22    D23
000F A 0010 A 0011 A 0012 A 0013 A 0002 A 0014 A 0015 A 0016 A 0017 A

D24    D25    D26    D3     D4     D5     D6     D7     D8     D9
0018 A 0019 A 001A A 0003 A 0004 A 0005 A 0006 A 0007 A 0008 A 0009 A

DEND   DETS   DM1    DM12   DM2    DM4    DM7    DPMD   E10    F100
337C A 0004 A FFFF A FFF4 A FFFE A FFFC A FFF9 A 0019 A 146C A 146D A

EADC   EADC1  EADC2  EADC3  EADC4  EBKD   EBKD1  EBKDE  EBUF   ECCU
110F A 111F A 1130 A 1132 A 113B A 1E00 A 1E00 A 1E0A A 31DF A 10C2 A

ECCU1  ECCUI  ECCUR  ECER   ECMF   ECOM   ECOM1  ECOM2  ECOMR  ED23
10CB A 10DB A 10D9 A 10BF A 31E6 A 1212 A 1227 A 123B A 123F A 31FD A

EDERR  EDET   EDET1  EDET2  EDET3  EDET32 EDETI  EDETM  EDFTR  EDETT
333C A 12B2 A 12B6 A 12C6 A 12D3 A 12DE A 12DC A 12DD A 12DB A 3202 A

EDIS   EDISH  EDLAY  EDMIT  EDNO   EDSAV  EDSV2  EDSVF  EDSVS  EEOC
1200 A 120F A 110E A 3201 A 31FE A 3243 A 3245 A 3247 A 3258 A 131B A

EERR   EERR0  EERR1  EERR2  EERR3  EERR4  EERR5  EERRF  EGIS   EGIS1
32DE A 32DF A 32E1 A 32E3 A 32E5 A 32E7 A 32E9 A 32DE A 1041 A 104D A
```

```
EGIS2  EGIS3  EGIS4  EGIS5  EHLT   EM10   EM24   EMAD   EMADE  EMADR
105E A 108B A 10A7 A 10AA A 1039 A 174D A 1752 A 10E0 A 10FF A 10F2 A

EMIC   EMIC1  EMIC2  EP1E   EP2E   EP3E   EPAN   EPAN1  EPANE  EPANE1
1282 A 1291 A 12A0 A 3297 A 3298 A 3299 A 1400 A 1426 A 1420 A 1422 A

EPBI   EPBI1  EPBI2  EPBI3  EPBIR  EPCD1  EPCD2  EPCD3  EPCF   EPCY
1726 A 173E A 1733 A 1726 A 1745 A 3276 A 3280 A 328A A 329C A 146A A

EPD1   EPD2   EPD3   EPDT   EPDVE  EPEF1  EPEF2  EPEF3  EPEOC  EPFI
3261 A 3268 A 326F A 31E5 A 329A A 32EB A 3306 A 3321 A 1321 A 1461 A

EPFI1  EPFIR  EPFT   EPIN1  EPIR   EPP1   EPP10  EPP11  EPP12  EPP13
1462 A 1469 A 3294 A 146B A 329D A 14C4 A 14F3 A 150C A 1525 A 153E A

EPP14  EPP15  EPP2   EPP20  EPP21  EPP22  EPP23  EPP24  EPP25  EPP3
155B A 1572 A 1578 A 15A7 A 15C0 A 15D9 A 15F2 A 160F A 1626 A 162C A

EPP30  EPP31  EPP32  EPP33  EPP34  EPP35  EPPM   EPPM1  EPPM2  EPPM3
165B A 1674 A 168D A 16A6 A 16C3 A 16DA A 146E A 1478 A 148B A 1499 A

EPPM4  EPPM5  EPPMR  EPST   EPUF   EPUP   EPUPS  ER123  ERE0   ERE1
14A7 A 14B5 A 14C2 A 31E4 A 329B A 16E0 A 171A A 31F9 A 1340 A 1349 A

ERE10  ERE11  ERE12  ERE2   ERE3   ERE4   ERE5   ERE6   ERE7   ERE8
139A A 13A3 A 13AC A 1352 A 135B A 1364 A 136D A 1376 A 137F A 1388 A

ERE9   ERN0   ERSI   ERTC   ERTC1  ERTCF  ERTS   ESAV   ESCH   ESDW
1391 A 31FA A 1000 A 108B A 1098 A 31DF A 110D A 3203 A 10F3 A 12E0 A

ESDW1  ESTK   ESTK1  ESV2   ESVF   ESVS   ESYA   ESYA1  ESYA2  ESYD
1301 A 1061 A 1078 A 3205 A 3207 A 3218 A 1279 A 127C A 1281 A 31E7 A

ESYN   ESYN1  ESYN2  ESYN3  ESYN4  ESYN4A ESYN5  ESYN6  ETCPY  ETCY
123F A 1257 A 125C A 1261 A 1267 A 1266 A 1272 A 1278 A 1023 A 31E1 A

ETEOC  ETEOC1 ETEOC2 ETIM   ETIM1  ETIMR  ETST   EVCVC  EZDB   FC1
1333 A 1339 A 133D A 12A3 A 12AD A 12B1 A 31E0 A 12DF A 101C A 0001 A

FC10   FC11   FC12   FC13   FC14   FC15   FC17   FC2    FC3    FC4
0010 A 0011 A 0012 A 0013 A 0014 A 0015 A 0017 A 0002 A 0003 A 0004 A

FC41   FC42   FC43   FC44   FC45   FC46   FC47   H0006  H0009  H000F
0041 A 0042 A 0043 A 0044 A 0045 A 0046 A 0047 A 1F81 A 174C A 12A1 A

H000F  H0010  H0030  H007F  H0090  H00F0  H0200  H0400  H0900  H0F00
1749 A 131A A 12A2 A 1F82 A 174B A 1748 A 00F5 A 14C3 A 174A A 1747 A

H0FFF  H1T1   H1T2   H2T1   H2T2   H3000  H3BFF  H3FFF  H7F00  H7FFF
1040 A 1C8E A 1C8F A 1C90 A 1C91 A 00F1 A 00F6 A 00F2 A 1F83 A 00F0 A
```

HEC00	HECCF	HF800	HFFEF	INEN	ISTK1	ISTK2	ISTK3	ISTK4	ISTK5
00F3 A	00F4 A	1318 A	1319 A	0001 A	3233 A	3234 A	3235 A	3236 A	3237 A
ISV	ISV0	ISV1	ISV2	ISV3	LAC	LBP	LBX	LDB	LDU
322F A	322F A	3230 A	3231 A	3232 A	0001 A	0001 A	0001 A	0001 A	0001 A
LEB	LER	LES	LPP	LPR	LTP	LVC	P10	P12	P14
0001 A	0001 A	0001 A	0001 A	0001 A	0001 A	0001 A	1BC4 A	1BD5 A	1BEF A
P14A	P18	P18A	P2	P20	P21	P3	P5	P6	P7
1BF1 A	1C02 A	1C04 A	1B36 A	1C10 A	1C16 A	1B55 A	1B65 A	1B7F A	1B9A A
P8	PBUF	PC1	PC15	PC2	PC3	PCIM	PEMAX	PERR	PERRF
1BAB A	311E A	1A5C A	1A5F A	1A73 A	1A84 A	1A4E A	312B A	32C4 A	32C4 A
PFREQ	PIP	PM01	PM02	PM03	PM04	PM05	PMAS	PMDT	PMHD
3126 A	311E A	1ACD A	1A14 A	1A15 A	1A20 A	1A1A A	1AE0 A	3127 A	1A34 A
PMIF	PMON	PMRET	PR0	PR1	PREQ	PRER8	PRER9	PRET	PRHD
3123 A	1A00 A	1A33 A	1AFE A	1B0B A	1AE6 A	1AF4 A	1AF8 A	1AED A	1AE4 A
PRLD	PSWCH	PSWFC	PSWFD	PT18A	PT2	PT20	PT3	PT4	PT5
1AE5 A	1A35 A	1A4C A	1A4D A	1B54 A	1A99 A	1B7D A	1A9B A	1AB2 A	1ABD A
PT6	PT7	PT8	PTAVT	PTCP	PTCS	PTEI	PTEM	PTFND	PTER
1AC2 A	1AC6 A	1ACD A	3128 A	3120 A	3121 A	3122 A	311F A	1ADC A	1A88 A
PTITE	PTPMT	PTRET	SEAD	SELF	SMAD	SMAD1	SMAD2	SMAD3	SMADE
3129 A	312A A	1A8F A	329F A	0002 A	1E0C A	1E0F A	1E14 A	1E1C A	1E19 A
SSAD	T0	T1	T2	T3	TAI1	TAI2	TBUF	TCDS	TCIW
329E A	182D A	182E A	182F A	1830 A	18D5 A	18D6 A	3171 A	3172 A	1918 A
TCIWC	TCOPY	TCTP	TD1	TD2	TD3	TDCH	TDIAL	TDSFL	TDWL
3171 A	18C9 A	31CB A	1880 A	18AA A	18AE A	18B4 A	1896 A	18B1 A	3178 A
TECI	TEMP	TEOC	TERR	TERR0	TERR1	TERRF	TEZ	TFST	TGZ
18D8 A	3177 A	3176 A	32D9 A	32DA A	32DC A	32D9 A	1844 A	3175 A	1852 A
THI	TIMX	TIN	TINT	TLOOP	TLZ	TMSG	TNCY	TND2	TNDC
18D4 A	18D7 A	3173 A	3174 A	18CA A	184D A	180E A	1866 A	18AC A	18C0 A
TNTP	TNXT	TOL	TPE	TPEX	TPM1	TPM2	TPM3	TPTAB	TPUE
31B7 A	1831 A	1859 A	1967 A	1800 A	317B A	318F A	31A3 A	192C A	1896 A
TRSET	TSCI	TSD2	TSD3	TSST	TSYNC	TSYNCE	TSYNCR	VC01	VC02
1829 A	18D3 A	1878 A	187F A	1818 A	1870 A	1888 A	1886 A	1C40 A	1C4D A
VC31	VC32	VC41	VC42	VCBT	VCBUS	VCC0	VCCP	VCDB	VCDHK
1C8A A	1C8B A	1C8C A	1C8D A	3130 A	1D92 A	1D15 A	1DI4 A	3151 A	1C92 A
VCDT	VCER	VCER1	VCER2	VCER3	VCER4	VCGTF	VCHD	VCHF	VCHFE
312F A	1DA2 A	1C72 A	1C76 A	1C7A A	1C7E A	1C6B A	1C82 A	1CEC A	1CBC A

```
VCK    VCK0   VCK1   VCK2   VCK21  VCK3   VCK5   VCK6   VCK7   VCK8
1CF6 A 1D0C A 1D10 A 1D26 A 1D27 A 1D33 A 1D42 A 1D5A A 1D5F A 1D61 A

VCK9   VCKFE  VCLAS  VCLD   VCLOG  VCM15  VCMA   VCMOF  VCMT   VCNK
1D77 A 1CED A 1C19 A 1C83 A 3165 A 1D22 A 1C84 A 1C86 A 1C85 A 1D8B A

VCRET  VCSDB  VCST1  VCST2  VCST3  VCVC   VCVNB  XBUF   XCCF   XCER
1C3E A 3000 A 1C87 A 1C88 A 1C89 A 3131 A 1D9B A 3119 A 1F66 A 1F43 A

XCFT   XCFTE  XCIWC  XCIWP  XERR   XERR0  XERR1  XERRF  XFO1   XFO2
1F85 A 1F96 A 311C A 311D A 32BF A 32C0 A 32C2 A 32BF A 1F5C A 1F72 A

XFOW   XIMX   XLOF1  XLOF2  XLON1  XLON2  XMCE   XOCW1  XOCW2  XRTC
1F4B A 1F96 A 1F9A A 1FA0 A 1F97 A 1F9D A 1F50 A 311A A 311B A 1F28 A

XSDC   XSDW   XWDT
1F36 A 3119 A 1F32 A

   DF2C    9B70
```

SNUMB = P833J, ACTIVITY # = 03, REPORT CODE = 74, RECORD COUNT = 000026

```
ORIGIN   DATE MODULE ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                SUBPROGRAMS INCLUDED IN DECK.

$       OPTION  FORTRAN,NOMAP
$       LIBRARY L1

                SUBPROGRAMS OBTAINED FROM USERS   LIBRARY   L1

                SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

                                RANGE                SIZE
ALLOCATED CORE     000000 THRU 047777       050000
   RELOCATABLE     003400 THRU 047777       044400
$       PRMFL   L1,R,R,USERLIB/UTILITY/BUTIL
$       FILE    10,X1R
$       FILE    11,X2R,40L
$       FILE    S1,X3R,44R

FCB AND BUFFER SPACE
   AVAILABLE       000101 THRU 003377       003277
   FILE CTRL BLKS  003174 THRU 003400       000205
   MAXIMUM BUFFER SPACE REQUIRED            002404

    20K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN     730517 F/8
001306 LOCATIONS REQUIRED FOR LOAD TABLE
EXECUTION PROGRAM ENTERED AT  047356   THROUGH  .FSETU
```

214	ABBUF										
		1078	1102	3395	4623						
215	ABDIN										
		1056	1227	1246	1277	1429	1437	2916	3000	3546	3666
216	ABERR										
		1409									
388	ABLSMU										
		NOT USED									
173	ABPAN										
		1688	1898								
337	ABPD1										
		2000	2067	2100	2133	2166	2201				
338	ABPD2										
		2240	2307	2340	2373	2406	2441				
339	ABPD3										
		2480	2547	2580	2613	2646	2681				
336	ABPPD										
		1860	3054	3673	4634	4649					
169	ABRTC										
		499	831								
8	ACO										
		431	448	449	454	469	472	474	484	485	492
		493	531	555	559	561	562	579	580	582	583
		585	586	593	594	595	596	597	598	599	600
		601	602	603	604	608	613	632	649	653	654
		655	657	658	660	663	664	680	681	684	705
		706	708	712	736	737	742	746	756	757	759
		760	762	783	784	785	786	795	796	797	798
		821	847	848	865	870	875	877	878	895	896
		898	899	901	902	908	913	918	937	941	943
		944	961	962	964	965	967	968	973	988	993
		1014	1015	1017	1021	1056	1071	1072	1076	1077	1080
		1081	1087	1088	1090	1091	1093	1094	1095	1100	1101
		1104	1105	1111	1112	1114	1115	1117	1118	1119	1122
		1135	1138	1140	1142	1144	1146	1149	1150	1151	1152
		1154	1155	1157	1162	1163	1169	1170	1176	1177	1185
		1186	1189	1190	1191	1192	1196	1197	1200	1201	1208
		1214	1227	1228	1232	1235	1237	1238	1240	1241	1246
		1247	1251	1254	1256	1257	1259	1260	1277	1282	1283
		1285	1286	1288	1289	1291	1301	1302	1308	1309	1333
		1334	1335	1337	1346	1347	1352	1353	1372	1374	1376
		1377	1378	1381	1383	1385	1388	1390	1392	1395	1397
		1399	1404	1405	1407	1409	1411	1413	1416	1418	1421
		1422	1424	1426	1429	1432	1434	1437	1440	1442	1461
		1462	1480	1483	1484	1485	1486	1490	1491	1492	1493
		1494	1495	1496	1497	1498	1510	1511	1532	1533	1535
		1536	1538	1543	1544	1546	1547	1549	1554	1555	1557
		1558	1560	1565	1566	1568	1569	1571	1576	1577	1579
		1580	1582	1587	1588	1590	1591	1593	1598	1599	1601
		1602	1604	1609	1610	1612	1613	1615	1620	1621	1623
		1624	1626	1631	1632	1634	1635	1637	1642	1643	1645
		1646	1648	1653	1654	1656	1657	1659	1664	1665	1667
		1668	1670	1691	1692	1693	1694	1695	1732	1751	1759
		1767	1775	1783	1798	1803	1808	1813	1835	1838	1860
		1861	1864	1867	1870	1871	1872	1874	1876	1887	1888
		1889	1890	1891	1916	1938	1959	1974	2001	2013	2014
		2024	2025	2026	2028	2032	2033	2035	2037	2040	2041

2047	2062	2063	2064	2068	2079	2080	2095	2096	2097
2101	2112	2113	2128	2129	2130	2134	2145	2146	2161
2162	2163	2167	2174	2181	2190	2197	2198	2202	2214
2215	2226	2227	2228	2241	2253	2254	2264	2265	2266
2268	2272	2273	2275	2277	2280	2281	2287	2302	2303
2304	2308	2319	2320	2335	2336	2337	2341	2352	2353
2368	2369	2370	2374	2385	2386	2401	2402	2403	2407
2414	2421	2436	2437	2438	2442	2454	2455	2466	2467
2468	2481	2493	2494	2504	2505	2506	2508	2512	2513
2515	2517	2520	2521	2527	2542	2543	2544	2548	2559
2560	2575	2576	2577	2581	2592	2593	2608	2609	2610
2614	2625	2626	2641	2642	2643	2647	2654	2661	2676
2677	2678	2682	2694	2695	2706	2707	2708	2799	2800
2801	2803	2804	2806	2807	2809	2810	2812	2813	2830
2843	2854	2888	2890	2896	2898	2916	2921	2922	2957
2958	2960	2962	2981	2982	2991	3000	3003	3010	3011
3015	3016	3021	3022	3023	3024	3026	3027	3029	3043
3045	3046	3049	3054	3086	3087	3090	3092	3100	3101
3103	3104	3111	3119	3120	3121	3123	3124	3126	3182
3183	3311	3313	3315	3318	3320	3322	3325	3330	3339
3341	3346	3347	3350	3355	3357	3359	3360	3363	3368
3370	3373	3375	3378	3379	3380	3395	3451	3452	3453
3455	3459	3460	3465	3472	3481	3485	3488	3489	3494
3495	3505	3506	3511	3514	3520	3522	3524	3525	3530
3546	3551	3553	3557	3563	3569	3574	3578	3581	3584
3594	3595	3596	3598	3599	3608	3610	3615	3618	3622
3623	3625	3626	3632	3634	3635	3636	3637	3638	3639
3640	3666	3685	3686	3690	3694	3695	3696	3698	3703
3709	3715	3717	3733	3734	3744	3745	3746	3747	3749
3751	3752	3753	3754	3756	3758	3759	3761	3771	3772
3773	3776	3777	3780	3782	3784	3788	3789	3790	3792
3794	3796	3797	3798	3810	3811	3812	3813	3814	3815
3816	3817	3819	3820	3821	3822	3823	3832	3833	3838
3840	3841	3843	3844	3845	3849	3851	3854	3855	3864
3866	3869	3870	3875	3876	3877	3878	3880	3882	3883
3884	3885	3887	3889	3890	3891	3892	3901	3902	3903
3904	3906	3908	3909	3910	3911	3913	3914	3916	3920
3924	3930	3931	3933	3934	3935	3937	3942	3943	3955
3956	3961	3963	3966	3968	3970	3975	3976	3989	3991
3993	3994	3996	3998	4003	4004	4006	4009	4010	4011
4015	4016	4021	4027	4032	4037	4038	4039	4041	4042
4043	4046	4047	4051	4052	4078	4085	4086	4087	4088
4089	4090	4091	4092	4093	4094	4095	4096	4097	4098
4100	4108	4111	4114	4115	4122	4138	4139	4142	4143
4145	4147	4149	4150	4152	4154	4156	4159	4160	4161
4164	4225	4226	4228	4229	4231	4232	4235	4236	4237
4238	4242	4243	4263	4264	4293	4300	4317	4324	4351
4352	4353	4358	4363	4366	4368	4418	4426	4427	4430
4433	4443	4444	4445	4450	4456	4459	4461	4482	4485
4492	4500	4503	4511	4522	4523	4529	4538	4550	4563
4564	4565	4566	4585	4597	4606	4625	4628	4630	4634
4669	4671	4673	4675	4677	4679	4681	4693	4695	4698
4703	4713	4715	4719	4720	4721	4722	4723	4724	4728
4729	4730	4731	4733	4737	4738	4740	4745	4747	4748
4749	4751	4752	4754	4758	4759	4761	4763	4764	4766
4771	4773	4774	4775	4777	4778	4780	4784	4785	4788
4792	4793	4794	4795	4796	4797	4867	4873	4879	4885

9 AC1

513	514	520	521	556	564	575	609	618	628
633	691	702	713	743	748	866	880	891	938
946	957	1000	1011	1022	1079	1081	1086	1103	1105
1110	1137	1138	1141	1142	1145	1146	1230	1233	1235
1249	1252	1254	1315	1316	1317	1322	1340	1378	1383
1385	1390	1392	1397	1399	1416	1418	1424	1426	1432
1434	1440	1442	1444	1505	1506	1508	1510	1513	1514
1515	1531	1542	1553	1564	1575	1586	1597	1608	1619
1630	1641	1652	1663	1714	1720	1726	1730	1731	1738
1749	1750	1757	1758	1765	1766	1773	1774	1781	1782
1834	1838	1881	1882	1900	1901	1904	1905	1921	1922
1925	1926	1943	1944	1947	1948	1964	1965	1968	1969
1980	1981	1983	1984	2005	2007	2015	2017	2021	2026
2045	2051	2054	2055	2072	2073	2081	2085	2086	2089
2105	2106	2114	2118	2119	2122	2138	2139	2147	2151
2152	2155	2171	2178	2187	2188	2189	2190	2206	2208
2216	2220	2245	2247	2255	2257	2261	2266	2285	2291
2294	2295	2312	2313	2321	2325	2326	2329	2345	2346
2354	2358	2359	2362	2378	2379	2387	2391	2392	2395
2411	2418	2427	2428	2429	2430	2446	2448	2456	2460
2485	2487	2495	2497	2501	2506	2525	2531	2534	2535
2552	2553	2561	2565	2566	2569	2585	2586	2594	2598
2599	2602	2618	2619	2627	2631	2632	2635	2651	2658
2667	2668	2669	2670	2686	2688	2696	2700	2724	2725
2729	2730	2735	2736	2740	2741	2746	2747	2751	2752
2757	2758	2761	2762	2765	2766	2770	2772	2775	2776
2779	2780	2787	2788	2792	2830	2831	2832	2835	2836
2837	2838	2839	2840	2843	2844	2845	2848	2849	2850
2851	2854	2855	2856	2859	2887	2888	2889	2890	2895
2896	2897	2898	2911	2913	2924	2946	2947	2948	2955
2977	2979	2980	2986	2996	3006	3007	3062	3063	3070
3071	3078	3079	3106	3127	3128	3130	3133	3137	3139
3143	3146	3181	3184	3401	3402	3405	3406	3408	3411
3416	3417	3420	3421	3423	3426	3476	3477	3490	3500
3501	3502	3507	3541	3543	3567	3568	3569	3572	3607
3662	3664	3673	3674	3679	3683	3724	3725	3728	3730
3775	3777	3778	3789	3797	3800	3848	3850	3921	3922
3925	3926	3939	3940	3958	3959	3968	3980	3982	3983
3985	3994	4008	4009	4014	4016	4017	4018	4031	4032
4033	4036	4037	4040	4102	4103	4105	4119	4132	4133
4173	4179	4185	4192	4240	4241	4254	4256	4257	4260
4261	4262	4275	4291	4294	4296	4309	4311	4315	4318
4320	4333	4335	4356	4359	4361	4364	4371	4372	4374
4375	4376	4378	4379	4382	4383	4416	4425	4431	4441
4442	4447	4448	4451	4453	4454	4457	4464	4465	4467
4468	4469	4471	4472	4475	4476	4479	4480	4483	4501
4510	4512	4530	4531	4547					

10 AC2

458	460	468	469	473	474	475	477	482	484
486	488	491	493	507	508	512	513	518	519
520	553	554	555	556	557	565	567	569	571
573	574	606	607	608	609	610	617	620	622
624	626	627	631	632	633	634	635	648	649
652	653	654	657	658	660	664	679	681	692
693	694	696	698	700	711	712	713	714	715
744	746	747	751	752	753	754	757	760	846

849	863	864	865	866	867	881	883	885	887
[illegible]	[illegible]	935	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	949	[illegible]
953	955	956	1001	1002	1003	1005	1007	1009	1020
1021	1022	1023	1024	1075	1076	1078	1079	1085	1086
1087	1090	1091	1093	1099	1100	1102	1103	1109	1110
1111	1114	1115	1117	1148	1150	1151	1154	1155	1157
1162	1163	1169	1170	1176	1177	1183	1185	1186	1190
1196	1197	1209	1214	1231	1234	1238	1250	1253	1257
1314	1317	1318	1322	1330	1331	1332	1333	1334	1337
1373	1374	1489	1490	1492	1493	1495	1496	1498	1504
1505	1506	1508	1511	1513	1515	1711	1717	1723	1729
1731	1732	1738	1748	1750	1751	1756	1758	1759	1764
1766	1767	1772	1774	1775	1780	1782	1783	1835	1839
1879	1882	1905	1926	1948	1969	2000	2001	2006	2007
2023	2024	2035	2048	2049	2067	2068	2100	2101	2133
2134	2166	2167	2173	2174	2182	2183	2184	2186	2187
2189	2190	2201	2202	2207	2208	2240	2241	2246	2247
2263	2264	2275	2288	2289	2307	2308	2340	2341	2373
2374	2406	2407	2413	2414	2422	2423	2424	2426	2427
2429	2430	2441	2442	2447	2448	2480	2481	2486	2487
2503	2504	2515	2528	2529	2547	2548	2580	2581	2613
2614	2646	2647	2653	2654	2662	2663	2664	2666	2667
2669	2670	2681	2682	2687	2688	2721	2722	2725	2727
2728	2730	2732	2733	2736	2738	2739	2741	2743	2744
2747	2749	2750	2752	2756	2757	2760	2761	2764	2765
2769	2770	2774	2775	2778	2779	2783	2788	2792	2800
2801	2804	2807	2810	2813	2829	2836	2838	2840	2842
2849	2851	2853	2859	2912	2913	2920	2945	2948	2949
2952	2953	2956	2958	2978	2979	2982	2990	3012	3014
3015	3020	3022	3023	3026	3027	3029	3044	3045	3046
3060	3063	3071	3079	3117	3119	3120	3123	3124	3126
3149	3154	3161	3165	3182	3187	3374	3375	3396	3411
3426	3458	3459	3460	3465	3466	3472	3473	3476	3494
3507	3513	3514	3523	3524	3542	3543	3552	3553	3559
3562	3563	3564	3565	3566	3567	3570	3571	3572	3573
3580	3581	3584	3590	3591	3595	3602	3603	3604	3607
3608	3609	3610	3614	3615	3624	3625	3631	3632	3635
3636	3638	3663	3664	3671	3672	3674	3713	3714	3715
3720	3721	3742	3749	3756	3759	3760	3761	3762	3770
3772	3780	3784	3785	3786	3787	3788	3791	3792	3794
3799	3800	3809	3811	3819	3821	3837	3840	3843	3845
3846	3847	3849	3850	3852	3853	3854	3855	3873	3880
3887	3890	3899	3906	3909	3914	3915	3916	3917	3920
3926	3929	3930	3932	3933	3941	3942	3943	3954	3955
3959	3960	3961	3963	3966	3969	3970	3979	3980	3981
3982	3984	3985	3986	3987	3988	3989	3991	3993	3995
3996	3997	3998	3999	4005	4006	4007	4008	4018	4021
4029	4031	4033	4034	4074	4075	4076	4086	4088	4090
4092	4094	4096	4098	4108	4115	4139	4143	4145	4147
4150	4152	4154	4156	4160	4225	4226	4228	4229	4231
4232	4235	4236	4237	4238	4240	4244	4260	4261	4265
4292	4293	4294	4297	4299	4301	4316	4317	4318	4321
4323	4325	4352	4353	4357	4358	4359	4362	4363	4364
4366	4368	4375	4378	4379	4382	4416	4417	4419	4425
4427	4430	4431	4432	4434	4442	4445	4447	4448	4450
4451	4453	4454	4456	4457	4459	4461	4468	4471	4472
4475	4480	4482	4483	4484	4486	4493	4500	4501	4502
4504	4510	4513	4515	4586	4591	4602	4608	4623	4625
4628	4630	4649	4653	4658	4669	4671	4673	4675	4677
4679	4681	4692	4693	4695	4698	4712	4713	4718	4719
4721	4722	4724	4727	4728	4730	4731	4733	4744	4745
4746	4747	4748	4751	4752	4754	4763	4770	4771	4772
4773	4774	4777	4778	4780	4790	4794	4797		

11 AC3

430	453	459	464	465	466	483	485	487	532
536	537	557	566	567	568	569	570	571	572
573	610	612	619	620	621	622	623	624	625
626	634	667	668	669	673	674	694	695	696
697	698	699	700	701	714	820	867	869	882
883	884	885	886	887	888	889	914	917	939
948	949	950	951	952	953	954	955	972	987
1003	1004	1005	1006	1007	1008	1009	1010	1023	1210
1211	1229	1230	1231	1232	1233	1234	1237	1240	1248
1249	1250	1251	1252	1253	1256	1259	1306	1308	1314
1323	1324	1326	1341	1342	1344	1347	1348	1350	1531
1532	1535	1536	1538	1542	1543	1546	1547	1549	1553
1554	1557	1558	1560	1564	1565	1568	1569	1571	1575
1576	1579	1580	1582	1586	1587	1590	1591	1593	1597
1598	1601	1602	1604	1608	1609	1612	1613	1615	1619
1620	1623	1624	1626	1630	1631	1634	1635	1637	1641
1642	1645	1646	1648	1652	1653	1656	1657	1659	1663
1664	1667	1668	1670	1698	1701	1704	1707	1713	1714
1719	1720	1725	1726	1734	1753	1761	1769	1777	1785
1800	1805	1810	1815	1822	1895	1910	1918	1932	1940
1953	1961	1976	2004	2005	2011	2016	2017	2020	2021
2028	2033	2036	2037	2038	2044	2045	2050	2051	2055
2060	2071	2072	2073	2077	2085	2088	2089	2093	2104
2105	2106	2110	2118	2121	2122	2126	2137	2138	2139
2143	2151	2154	2155	2159	2170	2171	2194	2205	2206
2212	2219	2220	2224	2244	2245	2251	2256	2257	2260
2261	2268	2273	2276	2277	2278	2284	2285	2290	2291
2295	2300	2311	2312	2313	2317	2325	2328	2329	2333
2344	2345	2346	2350	2358	2361	2362	2366	2377	2378
2379	2383	2391	2394	2395	2399	2410	2411	2434	2445
2446	2452	2459	2460	2464	2484	2485	2491	2496	2497
2500	2501	2508	2513	2516	2517	2518	2524	2525	2530
2531	2535	2540	2551	2552	2553	2557	2565	2568	2569
2573	2584	2585	2586	2590	2598	2601	2602	2606	2617
2618	2619	2623	2631	2634	2635	2639	2650	2651	2674
2685	2686	2692	2699	2700	2704	2723	2724	2729	2734
2735	2740	2745	2746	2751	2755	2758	2762	2766	2771
2772	2776	2780	2784	2789	2793	2799	2803	2806	2809
2812	2919	2920	2921	2928	2929	2930	2931	2942	2943
2949	2953	2985	2986	2989	2990	2991	2995	2996	3003
3009	3010	3013	3014	3098	3142	3143	3147	3148	3149
3150	3152	3153	3154	3155	3160	3163	3164	3165	3166
3183	3188	3452	3477	3487	3488	3489	3490	3495	3500
3502	3521	3522	3571	3573	3574	3591	3593	3594	3596
3597	3601	3603	3605	3612	3637	3678	3682	3698	3699
3700	3702	3707	3720	3721	3725	3728	3730	3734	3745
3747	3752	3754	3758	3776	3782	3785	3790	3810	3815
3817	3820	3833	3838	3841	3847	3848	3851	3853	3864
3866	3870	3876	3878	3883	3885	3889	3902	3904	3908
3913	3922	3924	3925	3931	3935	3939	3956	3958	3976
3983	3986	3997	4010	4011	4014	4017	4039	4065	4067
4068	4071	4073	4075	4078	4085	4087	4089	4091	4093
4095	4097	4100	4103	4106	4120	4123	4124	4126	4131
4133	4134	4135	4165	4166	4168	4174	4180	4186	4191
4284	4285	4287	4345	4346	4348	4350	4356	4361	4449
4455	4495	4512	4519	4523	4532	4533	4535	4539	4540

4602	4607	4608	4622	4626	4632	4650	4655	4660	4691
[illegible]	[illegible]	4757	.758	[illegible]	[illegible]	[illegible]	478[illegible]	.784	[illegible].35
4787	4788	4791	4792	4793	4795				

396	ACCU						
		NOT USED					
768	ACCUI						
		736					
770	ACDISP						
		763					
389	ACMU						
		453					
170	ACOTF						
		828					
771	ACRAM						
		759					
315	ACSAV						
		NOT USED					
316	ACSV2						
		935	936	1020	1024		
317	ACSVF						
		944	947	1014			
318	ACSVS						
		965	968	993	1001		
330	ACTST						
		3044	3542	3663			
772	ACWAP						
		744	751				
379	ADEND						
		477	783	4563			
392	ADMU						
		NOT USED					
358	AE100						
		2183	2423	2663			
359	AE10						
		2048	2288	2528			
156	AEADC						
		637	721	807	1060	1742	3031
145	AEBKD						
		536					
285	AEBUF						
		468	648	679			
143	AECCU						
		438					
773	AECCUR						
		756					
146	AECER						
		661					
293	AECMF						
		1072	1095	1119	1122	4715	
149	AECOM						
		833					
300	AED23						
		1241	1373	3111			
304	AEDERR						
		1332					

155	AEDET										
		837									
303	AEDETT										
		1323	1326	1330	1340	1341	1344				
148	AEDIS										
		832									
302	AEDMIT										
		1288	1291	1301	1353						
301	AEDNO										
		1229									
326	AEDSV2										
		863	864								
327	AEDSVF										
		878	881								
328	AEDSVS										
		899	902								
153	AEEOC										
		3008									
291	AEERR										
		652	1085	1109	1148	1324	1342	1404	1421		
142	AEGIS										
		133									
183	AEM10										
		2019	2087	2120	2153	2218	2259	2327	2360	2393	2458
		2499	2567	2600	2633	2698					
184	AEM24										
		2046	2286	2526							
144	AEMAD										
		442									
151	AEMIC										
		835									
361	AEP1E										
		1692	1798	1888	1916	2014	2041	2063	2080	2096	2113
		2129	2146	2162	2197	2215	2227				
362	AEP2E										
		1693	1803	1889	1938	2254	2281	2303	2320	2336	2353
		2369	2386	2402	2437	2455	2467				
363	AEP3E										
		1694	1808	1890	1959	2494	2521	2543	2560	2576	2593
		2609	2626	2642	2677	2695	2707				
174	AEPAN										
		496									
180	AEPBI2										
		2168	2408	2648							
181	AEPBI3										
		2002	2069	2102	2135	2203	2242	2309	2342	2375	2443
		2482	2549	2582	2615	2683					
345	AEPCD1										
		2020	2036	2050	2088	2121	2154	2173	2186	2219	2723
		2756	2769	2784							
346	AEPCD2										
		2260	2276	2290	2328	2361	2394	2413	2426	2459	2734
		2760	2774	2789							

347	AEPCD3										
		2500	2516	2530	2568	2601	2634	2653	2666	2699	2745
		2764	2778	2793							
352	AEPCF										
		1871	1876	1879	1901	1922	1944	1965	1984		
356	AEPCY										
		2006	2015	2246	2255	2486	2495				
341	AEPD1										
		2004	2016	2044	2071	2104	2137	2170	2205		
342	AEPD2										
		2244	2256	2284	2311	2344	2377	2410	2445		
343	AEPD3										
		2484	2496	2524	2551	2584	2617	2650	2685		
290	AEPDT										
		NOT USED									
364	AEPDVE										
		1695	1813	1891	1974	2064	2097	2130	2163	2198	2228
		2304	2337	2370	2403	2438	2468	2544	2577	2610	2643
		2678	2708								
366	AEPEF1										
		1734	1753	2011	2038	2060	2077	2093	2110	2126	2143
		2159	2194	2212	2224						
367	AEPEF2										
		1761	1769	2251	2278	2300	2317	2333	2350	2366	2383
		2399	2434	2452	2464						
368	AEPEF3										
		1777	1785	2491	2518	2540	2557	2573	2590	2606	2623
		2639	2674	2692	2704						
349	AEPFT										
		1713	1719	1725	2023	2263	2503				
357	AEPIN1										
		2207	2216	2447	2456	2687	2696				
353	AEPIR										
		1867	1874								
176	AEPP1										
		1789	1913								
177	AEPP2										
		1792	1935								
178	AEPP3										
		1795	1956								
175	AEPPM										
		842									
289	AEPST										
		1192	1201	1381	2956						
351	AEPUF										
		1864	1981	3092	3101						
179	AEPUP										
		1820	3096								
297	AER123										
		1135	1140	1144	1149	1260	1372				
305	AERAM										
		449	980								

186	AERE0	1739	4549					
196	AERE10	2111	2351	2591				
197	AERE11	2144	2384	2624				
198	AERE12	2213	2453	2693				
187	AERE1	2061	2301	2541	4175			
188	AERE2	2094	2334	2574	4181			
189	AERE3	2127	2367	2607	4187			
190	AERE4	1343	2160	2400	2640	4193		
191	AERE5	1325	2195	2435	2675	4121		
192	AERE6	2225	2465	2705	4107			
193	AERE7	4110						
194	AERE8	2012	2039	2252	2279	2492	2519	3680
195	AERE9	2078	2318	2558	3684			
298	AERNO	1248						
141	AERSI	130						
310	AESAV	NOT USED						
147	AESCH	671						
154	AESDW	841	906					
311	AESV2	553	554	711	715			
312	AESVF	562	565	705				
313	AESVS	583	586	684	692			
295	AESYD	1209	2783					
150	AESYN	834						
287	AETCY	491	1282	1285	1504	3541	3662	
152	AETIM	836						
288	AETST	1286	2952					

321	AISTK1										
		594									
322	AISTK2										
		597									
323	AISTK3										
		600									
324	AISTK4										
		603	617								
320	AISV2										
		606	607	631	635						
387	ALDU										
		4626	4691	4702							
225	APBUF										
		NOT USED									
161	APCIM										
		3619	3627								
234	APEMAX										
		2771	3714	3717	3791	3941	3960	3988	4005	4027	
235	APERR										
		1413	3678	3682	4106	4120	4174	4180	4186	4191	4548
232	APFREQ										
		1480	1486	4043							
226	APIP										
		1388	3049	3090	3325	3368	3380	3551	3640	3686	3690
		3700	4052								
233	APMDT										
		3318	3322	4042							
231	APMIF										
		1489	3370	3374	3379	3557	3578	3614	3624	3631	3742
		3770	3809	3837	3873	3899					
157	APMON										
		838									
400	APP1U										
		1698	1800	1910	1918	4650					
401	APP2U										
		1701	1805	1932	1940	4655					
402	APP3U										
		1704	1810	1953	1961	4660					
399	APPDU										
		1707	1815	1822	1895	1976	3098	4632			
159	APREQ										
		3344									
236	APTAVT										
		3451	3501	3506	3521	3530	3612				
228	APTCP										
		3762	3787	3799	3813	3877	3917	3932	3954	3969	3979
		3984	3995								
229	APTCS										
		3746	3773	3816	3884	3903	4007				
230	APTEI										
		3753	3778	3823	3892	3911	4029				

```
227  APTEM
                3453  3520  3564  3570  3593  3609  4004  4015
160  APTER
                3333
237  APTITE
                3455  3485  3511  3618  3626
239  APTPMT
                3481  3487  3568  3623
394  ARTC
                 430   532   612   820   869   972   987
393  ASDB
            NOT USED
391  ASDU
                 917  4697
375  ASEAD
                 786   798  4566  4586
206  ASMAD
                 788   800  4568
374  ASSAD
                 784   796  4564  4587  4594
273  ATAI1
                1730  1757  1773  2721  2732  2743
274  ATAI2
                1749  1765  1781  2727  2738  2749
250  ATBUF
                4712  4744  4770  4790
252  ATCDS
                1210  2946  3060  3106  3127  3142  3146  3147  3152  3163
                3565  3671  3713
278  ATCIW
                 519  2919  3012
251  ATCIWC
                 521  2922  3016
163  ATCOPY
                3151  3156  3162  3167
390  ATCP
                 914  4622
265  ATCTP
                 512  2989  3009  3150  3160  3466  3559  3760  3812  3822
                3891  3910  3915
165  ATDIAL
                 504
258  ATDWL
                2755  2945
276  ATECI
                3702
257  ATEMP
            NOT USED
256  ATEOC
            NOT USED
266  ATERR
                1411  3020  3117
```

```
 255  ATFST
                1462  1484  2942  2960  2962  3087  3104  4047
 272  ATHI
                 507   518  2912  2995
 275  ATIMX
                2985
 253  ATIN
                 508  2911  2978  2980  3007  3013  3021  3373  3458  3513
                3580  3724  3832  3921  4038
 254  ATINT
                 514  2928  2930  3011  3523  3525  3590  3597  3599  3602
                3604  3733  3869  3929  3934  3940  3975  3981  3987  3999
                4036  4040
 164  ATNCY
                1471
 264  ATNTP
                3155  3161  3166  3473  3562  3634  3744  3751  3775  3786
                3814  3846  3852  3875  3882  3901  4034
 281  ATPE
                 488
 162  ATPEX
                 839
 261  ATPM1
                 483  1711  1729  1748  2722  2728
 262  ATPM2
                1717  1756  1764  2733  2739
 263  ATPM3
                1723  1772  1780  2744  2750
 259  ATPP
                3148  3153  3164  3566
 280  ATPTAB
                 482
 271  ATSCI
              NOT USED
 241  AVCBT
                1395  3311  3315  4531
4409  AVCBUS
                4336
 245  AVCDB
                4076
 244  AVCDT
                3330  3341  3346  3357  3359  3552  3679  3683  3694  4065
                4105  4119  4132  4173  4179  4185  4192  4547
4411  AVCER
                4424  4439  4491  4509
4525  AVCK
                4524
 158  AVCLAS
                3331  3342  3358
 243  AVCLOG
                4123  4165  4284  4345  4532  4539
```

246	AVCSDB	473	3396	4074							
242	AVCVC	1306	1315	4134							
4410	AVCVNB	4307	4310	4313	4331	4334	4337				
220	AXBUF	908	913	918	1075	1099	1444	4692	4718	4727	4760
		4787	4791								
4810	AXCFT	4757	4783								
221	AXERR	1407	4746	4772							
201	AXLOF1	1699	1702	1705	1823	1911	1933	1954	3099		
203	AXLOF2	1708	1896								
200	AXLON1	1801	1806	1811	1919	1941	1962				
202	AXLON2	1816	1977								
166	AXRTC	830									
168	AXSDC	524	840								
167	AXWDT	1125	1740								
4916	BBUF	214									
4918	BCR1	NOT USED									
4919	BCR2	NOT USED									
4917	BDIN	215									
5164	BERR	216									

```
                216
5165  BERRF
            NOT USED
4649  BPAN
                173
4950  BPD1
                337
4951  BPD2
                338
4952  BPD3
                339
4668  BPIN
            4651  4656  4661
4939  BPPD
                336
```

4622	BRTC										
		169									
91	C0										
		431	454	472	582	613	680	742	762	821	870
		898	964	973	988	1071	1189	1200	1346	1352	1483
		1485	1514	1691	1834	1887	1983	2032	2047	2178	2181
		2272	2287	2418	2421	2512	2527	2658	2661	2829	2842
		2853	2957	2981	3006	3043	3086	3100	3103	3378	3401
		3416	3505	3598	3622	3639	3685	3771	3844	3937	4046
		4051	4102	4138	4242	4263	4351	4426	4443	4444	4479
		4492	4511	4529	4703	4796	4867	4873	4879	4885	
102	C125										
		4114									
100	C15										
		NOT USED									
92	C1										
		460	475	486	487	531	579	585	655	663	747
		849	895	901	961	967	1088	1094	1112	1152	1191
		1283	1289	1316	1335	1348	1461	1533	1544	1555	1566
		1577	1588	1599	1610	1621	1632	1643	1654	1665	1839
		1900	1904	1980	2013	2025	2040	2062	2079	2086	2095
		2112	2119	2128	2145	2152	2161	2188	2196	2214	2226
		2253	2265	2280	2302	2319	2326	2335	2352	2359	2368
		2385	2392	2401	2428	2436	2454	2466	2493	2505	2520
		2542	2559	2566	2575	2592	2599	2608	2625	2632	2641
		2668	2676	2694	2706	2955	2977	3024	3062	3121	3139
		3184	3187	3188	3339	3405	3420	3601	3605	3699	3796
		3798	4111	4122	4164	4441	4538	4550	4590	4597	4607
		4749	4775								
93	C2										
		1118	1137	1921	1925	3070	3133	3347	3355	3360	
94	C3										
		618	1943	1947	3078	3137					
101	C48										
		743									
95	C4										
		564	627	691	880	946	1000	1141	1964	1968	3707
96	C5										
		NOT USED									
97	C6										
		1208	2787	4522							
98	C7										
		4653	4658								
99	C8										
		1145									
108	CCB0										
		581	897	963	4598						
109	CCB1										
		NOT USED									
116	CCCF										
		826									

113	CCCU										
		733									
106	CCEZ										
		650	685	994	1082	1106	1136	1139	1143	1147	1164
		1171	1178	1382	1389	1396	1414	1423	1733	1752	1760
		1768	1776	1784	1799	1804	1809	1814	1836	1877	1917
		1939	1960	1975	2992	3312	3319	3326	3369	3371	3482
		3531	3576	3616	4079	4739	4765				
110	CCNZ										
		1123	1236	1239	1255	1258	1406	1408	1410	1412	1481
		1862	1868	3050	3091	3093	3332	3343	3691	4104	4109
		4714	4716								
114	CCPH										
		528									
111	CCSF										
		429	551								
112	CCZN										
		1837	2027	2175	2267	2415	2507	2655	2993	3372	3456
		3512	3558	3575	3579	3706	3957	3977	4030		
107	CCZP										
		1512	3936	4497	4517						
84	CM100										
		NOT USED									
87	CM10										
		2081	2114	2147	2321	2354	2387	2561	2594	2627	
83	CM110										
		NOT USED									
86	CM16										
		458									
90	CM1										
		464	575	628	667	702	748	891	957	1011	1211
		1870	1881	2947	3313	3320	3703	4071	4131	4585	4606
85	CM20										
		3181									
89	CM2										
		3709									
88	CM6										
		760									
948	COTF1										
		958									
967	COTF2										
		963									
971	COTF3										
		966									
997	COTF4										
		994									
1002	COTF5										
		1012									
934	COTF										
		170									

5077	CSAV										
		315									
5080	CSV2										
		316									
5082	CSVF										
		317									
5084	CSVS										
		318									
5109	CTST										
		330									
46	DO										
		469	539	649	658	659	660	664	681	700	765
		812	852	912	1009	1026	1058	1091	1093	1115	1117
		1127	1155	1157	1158	1185	1204	1217	1230	1232	1237
		1240	1249	1251	1256	1259	1263	1293	1334	1337	1355
		1446	1476	1490	1492	1500	1505	1515	1517	1536	1537
		1538	1539	1547	1549	1550	1558	1560	1561	1569	1571
		1572	1580	1582	1583	1591	1593	1594	1602	1604	1605
		1613	1615	1616	1624	1626	1627	1635	1637	1638	1646
		1648	1649	1657	1659	1660	1668	1670	1671	1714	1732
		1751	1759	1767	1775	1783	1825	1835	1843	1986	2001
		2005	2007	2017	2024	2035	2045	2049	2073	2085	2106
		2118	2139	2151	2184	2187	2208	2230	2241	2245	2247
		2257	2275	2285	2289	2313	2325	2346	2358	2379	2391
		2424	2427	2448	2470	2481	2485	2487	2497	2515	2525
		2529	2553	2565	2586	2598	2619	2631	2664	2667	2688
		2710	2724	2735	2746	2758	2796	2800	2815	2862	2891
		2899	2923	2927	2933	2963	3017	3027	3029	3045	3124
		3125	3126	3149	3154	3157	3165	3168	3186	3190	3375
		3381	3418	3422	3427	3459	3460	3476	3488	3490	3494
		3495	3500	3502	3507	3514	3522	3524	3533	3549	3553
		3567	3571	3573	3581	3584	3591	3594	3595	3596	3608
		3610	3636	3638	3642	3664	3669	3674	3676	3681	3687
		3715	3720	3728	3730	3756	3758	3780	3782	3785	3792
		3794	3800	3810	3821	3838	3843	3847	3864	3866	3876
		3878	3890	3909	3913	3920	3922	3926	3930	3933	3942
		3943	3961	3963	3966	3982	3989	3991	3993	4006	4049
		4053	4087	4088	4112	4115	4125	4126	4128	4135	4150
		4160	4167	4170	4231	4235	4534	4537	4541	4544	4551
		4604	4625	4635	4663	4683	4698	4700	4705	4713	4725
		4734	4745	4752	4753	4754	4755	4763	4771	4778	4780
		4781	4794	4797	4798	4869	4875	4881	4887		
56	D10										
		1384	1575	1647	4363	4432	4456	4482	4502	4542	
73	D116										
		848									
74	D118										
		847									
57	D11										
		1587	1590	1658	3055	4364	4378	4416	4457	4471	4483
		4540									

58 D12

753 1586 1669 3398 4352 4427 4445 4480 4493 4515

59 D13

1598 1601 3413 4425 4442

60 D14

754 1278 1597 2917 3001 3547 3667 4248 4269 4305
4329 4422 4437 4489 4507

61 D15

1057 1438 1609 1612 2961 3411 3471 3486 3561 3633
3748 3755 3779 3783 3818 3839 3842 3879 3886 3905
4353 4513 4732 4786

62 D16

1608 4510

63 D17

580 896 962 1620 1623

64 D18

1619

65 D19

1631 1634

47 D1

557 567 610 620 634 654 657 698 714 867
883 909 939 949 1007 1023 1076 1079 1092 1162
1231 1233 1250 1252 1331 1333 1441 1493 1495 1506
1508 1532 1535 1548 1720 2028 2033 2068 2072 2174
2182 2189 2190 2264 2268 2273 2308 2312 2414 2422
2429 2430 2508 2513 2548 2552 2654 2662 2669 2670
2729 2740 2751 2762 2772 2804 2837 2848 2861 2887
2897 2959 3028 3046 3066 3120 3123 3463 3498 3532
3543 3625 3632 3745 3747 3759 3784 3811 3815 3817
3840 3880 3883 3885 3887 3902 3904 3914 4089 4090
4124 4143 4301 4325 4419 4434 4449 4610 4628 4669
4693 4719 4721 4728 4730 4748 4751 4761 4779

66 D20

1630

67 D21

1642 1645

68 D22

1641

69 D23

1653 1656

70 D24

1652

71 D25

1664 1667

72 D26

1663

48 D2

493 569 622 653 696 885 951 1005 1100 1103
1116 1169 1234 1253 1433 1496 1498 1511 1513 1531
1559 1726 2101 2105 2220 2341 2345 2460 2504 2581
2585 2700 2766 2770 2775 2776 2779 2807 2835 2850
2889 3119 3615 3721 3725 3749 3752 3754 3772 3776
3819 3820 3833 3841 3845 3853 3889 3906 3908 4039
4091 4092 4152 4168 4226 4237 4240 4261 4448 4455
4630 4671 4695 4722 4724 4731 4733 4747 4788

49	D3										
		571	624	694	757	887	953	1003	1087	1090	1156
		1176	1543	1546	1570	2021	2037	2054	2134	2138	2261
		2277	2294	2374	2378	2501	2517	2534	2614	2618	2780
		2799	2810	2839	2895	3023	3026	3734	3790	3848	3851
		3870	3924	3925	3931	3935	3939	3956	3958	3976	3983
		3986	3997	4010	4011	4014	4017	4073	4093	4094	4145
		4166	4292	4316	4361	4362	4673	4774	4777	4792	4795
50	D4										
		441	573	574	626	889	890	955	956	1039	1086
		1196	1425	1542	1581	2089	2167	2171	2329	2407	2411
		2569	2647	2651	2803	2813	3022	3112	4095	4096	4154
		4225	4229	4241	4262	4287	4356	4447	4454	4573	4603
		4675	4773	4793							
51	D5										
		444	1044	1111	1114	1163	1170	1177	1183	1186	1190
		1197	1417	1554	1557	1592	2122	2202	2206	2362	2442
		2446	2602	2682	2686	2801	2806	3074	3114	3135	4085
		4086	4108	4139	4147	4285	4297	4321	4357	4495	4677
52	D6										
		445	1048	1110	1376	1398	1430	1553	1603	2155	2395
		2635	2809	4097	4098	4156	4228	4232	4236	4260	4348
		4453	4512	4679							
53	D7										
		446	1052	1151	1154	1391	1565	1568	1614	2055	2295
		2535	2812	3426	4078	4100	4103	4238	4244	4265	4346
		4486	4504	4681							
54	D8										
		1150	1564	1625	2051	2291	2531	2757	2761	2765	4293
		4299	4317	4323	4358	4366	4368	4417	4430	4450	4459
		4461	4484	4500	4535	4738					
55	D9										
		1576	1579	1636	3082	4294	4318	4359	4375	4379	4382
		4431	4451	4468	4472	4475	4501	4533			
5258	DEND										
		379									
76	DETS										
		1361	3384	3651	4199						
41	DM12										
		746									
45	DM1										
		556	609	633	713	866	938	1022			
44	DM2										
		555	608	632	712	865	937	1021			
43	DM4										
		693	1002								
42	DM7										
		752									
78	DPMD										
		4041	4530								

1849	E100	358		
1848	E10	359		
882	EADC1	892		
901	EADC2	897		
904	EADC3	900		
917	EADC4	911		
862	EADC	156		
4563	EBKD1	4574		
4560	EBKD	145		
4578	EBKDE	4569	4575	
5025	EBUF	285		
746	ECCU1	749		
730	ECCU	143		
769	ECCUI	737		
765	ECCUR	734	739	773
720	ECER	146		
5036	ECMF	293		
1098	ECOM1	1082		
1122	ECOM2	1106		
1070	ECOM	149		
1127	ECOMR	1120	1123	
5054	ED23	300		
5253	EDERR	304		
1308	EDET1	1351		
1330	EDET2	1319		
1360	EDET32	1318		

```
1346  EDET3
                1311  1327
1300  EDET
                155
1358  EDETI
                1302
1359  EDETM
                1309
1355  EDETR
                1304
5060  EDETT
                303
1036  EDIS
                148
1060  EDISH
                1041  1046  1050  1054
 854  EDLAY
                846
5059  EDMIT
                302
5055  EDNO
                301
5099  EDSAV
                NOT USED
5102  EDSV2
                326
5104  EDSVF
                327
5106  EDSVS
                328
1460  EEOC
                153
5218  EERR0
                NOT USED
5219  EERR1
                NOT USED
5220  EERR2
                NOT USED
5221  EERR3
                NOT USED
5222  EERR4
                NOT USED
5223  EERR5
                NOT USED
5208  EERR
                291
5209  EERRF
                NOT USED
 566  EGIS1
                576
 585  EGIS2
                581
```

```
 643  EGIS3
                584   587
 688  EGIS4
                685
 693  EGIS5
                703
 550  EGIS
                142
 528  EHLT
            NOT USED
2886  EM10
                183
2894  EM24
                184
 779  EMAD
                144
 807  EMADE
                789   801
 812  EMADR
                805
1243  EMIC1
               1236  1239
1262  EMIC2
               1255  1258
1224  EMIC
                151
5140  EP1E
                361
5141  EP2E
                362
5142  EP3E
                363
1747  EPAN1
               1733
1685  EPAN
                174
1740  EPANE1
               1817
1737  EPANE
               1754  1762  1770  1778  1786
2853  EPBI1
            NOT USED
2842  EPBI2
                180
2829  EPBI3
                181
2827  EPBI
            NOT USED
2861  EPBIR
               2834  2847  2858
5122  EPCD1
                345
```

5132	EPCD2	346		
5133	EPCD3	347		
5146	EPCF	352		
1846	EPCY	356		
5117	EPD1	341		
5118	EPD2	342		
5119	EPD3	343		
5034	EPDT	290		
5143	EPDVE	364		
5243	EPEF1	366		
5246	EPEF2	367		
5249	EPEF3	368		
1479	EPEOC	1468		
1835	EPFI1	1841		
1833	EPFI	1712	1718	1724
1843	EPFIR	1837		
5136	EPFT	349		
1847	EPIN1	357		
5147	EPIR	353		
2059	EPP10	2003		
2093	EPP11	2070		
2126	EPP12	2103		
2159	EPP13	2136		
2194	EPP14	2169		
2224	EPP15	2204		
1997	EPP1	176		

2299 EPP20
2243
2333 EPP21
2310
2366 EPP22
2343
2399 EPP23
2376
2434 EPP24
2409
2464 EPP25
2444
2237 EPP2
177
2539 EPP30
2483
2573 EPP31
2550
2606 EPP32
2583
2639 EPP33
2616
2674 EPP34
2649
2704 EPP35
2684
2477 EPP3
178
1874 EPPM1
1865
1904 EPPM2
1884
1925 EPPM3
1907
1947 EPPM4
1928
1968 EPPM5
1950
1859 EPPM
175
1986 EPPMR
1877 1902 1923 1945 1966 1971
5033 EPST
289
5145 EPUF
351
2717 EPUP
179
2799 EPUPS
2785 2790 2794
5049 ER123
297

```
1530  ERE0
                186
1640  ERE10
                196
1651  ERE11
                197
1662  ERE12
                198
1541  ERE1
                187
1552  ERE2
                188
1563  ERE3
                189
1574  ERE4
                190
1585  ERE5
                191
1596  ERE6
                192
1607  ERE7
                193
1618  ERE8
                194
1629  ERE9
                195
5050  ERNO
                298
 428  ERSI
                141
 663  ERTC1
                650
 648  ERTC
            NOT USED
5026  ERTCF
            NOT USED
 852  ERTS
                844
5067  ESAV
                310
 819  ESCH
                147
1415  ESDW1
            1406  1408  1410  1412
1369  ESDW
                154
 619  ESTK1
                629
 592  ESTK
                429   551
5070  ESV2
                311
```

5072	ESVF	312				
5074	ESVS	313				
1211	ESYA1	1215				
1216	ESYA2	1213				
1206	ESYA	1161	1168	1175	1181	
5037	ESYD	295				
1160	ESYN1	1139				
1167	ESYN2	1143				
1174	ESYN3	1147				
1182	ESYN4	1165	1172	1179		
1181	ESYN4A	1136				
1195	ESYN5	1188				
1203	ESYN6	1164	1171	1178	1193	1199
1134	ESYN	150				
484	ETCPY	489				
5030	ETCY	287				
1510	ETEOC1	1507				
1514	ETEOC2	1509	1512			
1503	ETEOC	1474				
1288	ETIM1	1280				
1274	ETIM	152				
1293	ETIMR	NOT USED				
5028	ETST	288				
1361	EVCVC	1350				
474	EZDB	478				
18	FC10	4678				

19	FC11	4668					
20	FC12	4670					
21	FC13	4672					
22	FC14	4674					
23	FC15	4676					
30	FC17	4680					
13	FC1	919	4694	4699			
14	FC2	4696					
15	FC3	4624	4627	4633			
32	FC41	433	455	822	872	975	4704
33	FC42	533	989				
34	FC43	432	614	871	915	974	
35	FC44	4868					
36	FC45	4874					
37	FC46	4880					
38	FC47	4886					
16	FC4	4629					
4801	H0006	NOT USED					
2870	H0009	2856					
1266	H000E	1247					
2867	H000F	2855					
1451	H0010	1422					
1267	H0030	1228					
4802	H007F	4764					
2869	H0090	2845					
2866	H00F0	2844					

415	H0200	448					
1989	H0400	1861	1872				
2868	H0900	2832					
2865	H0F00	2831					
543	H0FFF	492					
4214	H1T1	4309					
4215	H1T2	4311					
4216	H2T1	4333					
4217	H2T2	4335					
410	H3000	NOT USED					
416	H3BFF	4565					
411	H3FFF	785	1077	1080	1101	1104	
4803	H7F00	4737					
409	H7FFF	1491	1494	1497	4720	4723	4729
413	HEC00	795					
414	HECCF	797					
1449	HF800	1377					
1450	HFFEF	1405					
119	INEN	435	615	824	873		
5092	ISTK1	321					
5093	ISTK2	322					
5094	ISTK3	323					
5095	ISTK4	324					
5096	ISTK5	NOT USED					
5088	ISV0	NOT USED					
5089	ISV1	NOT USED					

5090	ISV2	
		320
5091	ISV3	
		NOT USED
5087	ISV	
		NOT USED
5261	LAC	
		4
5262	LBP	
		125 137 209 308 333 371 382 405
5270	LBX	
		4617 4640 4687 4709 4807 4859
5271	LDB	
		4894 5064 5112 5150 5159
5272	LDU	
		NOT USED
5269	LEB	
		4557 4582
5263	LER	
		423 547 727 776 816 859 929 1521
5264	LFS	
		1033 1067 1131 1221 1271 1297 1366 1456
5265	LPP	
		1679 1853 1994 2234 2474 2714 2819 2874
5267	LPR	
		3298 3387 3435 3536 3646 3655 3739 3767 3807 3828
		3861 3896 3949
5266	LTP	
		2906 2937 2972 3038 3172 3194
5268	LVC	
		4059 4197 4220 4280 4340 4388 4414 4528
3952	P10	
		3824 3893
3973	P12	
		NOT USED
4001	P14	
		3957 3977
4004	P14A	
		3945 3964 3992
4025	P18	
		3962 3971 3990 4000 4022
4029	P18A	
		3803 3944 4019
4041	P20	
		3857
4051	P21	
		4030
3768	P2	
		3732 3735
3808	P3	
		3726

```
3830  P5
                3722
3862  P6
                3834
3897  P7
                3867  3872
3918  P8
                3763
4963  PEUF
                 225
3475  PC15
                3467
3469  PC1
                3456
3504  PC2
                3482  3491
3530  PC3
                3512  3517
3437  PCIM
                 161
4977  PEMAX
                 234
5178  PERR
                 235
5179  PERRF
            NOT USED
4970  PFREQ
                 232
4964  PIP
                 226
3325  PM01
                3319
3338  PM02
                3323  3326  3332
3340  PM03
                3352
3356  PM04
                3365
3346  PM05
                3343
3647  PMAS
                3672
4971  PMDT
                 233
3384  PMHD
                3350  3363
4969  PMIF
                 231
3304  PMON
                 157
3381  PMRET
                3369  3372  3377
```

3690	PRO			
		3675		
3706	PR1			
		3710		
3659	PREQ			
		159		
3678	PRER8			
		3697		
3682	PRER9			
		3706		
3669	PRET			
		3691		
3651	PRHD			
		3695		
3652	PRLD			
		3696		
3388	PSWCH			
		3307		
3430	PSWFC			
		3402	3406	3408
3431	PSWFD			
		3417	3421	3423
3803	PT18A			
		3802		
3857	PT20			
		3856		
3561	PT2			
		3558		
3563	PT3			
		3560		
3590	PT4			
		3585		
3601	PT5			
		3592		
3607	PT6			
		3579	3583	
3611	PT7			

3611	PT7					
		3600				
3622	PT8					
		3616				
4972	PTAVT					
		236				
4966	PTCP					
		228				
4967	PTCS					
		229				
4968	PTEI					
		230				
4965	PTEM					
		227				
3639	PTEND					
		3575	3576	3586	3620	3628

3538	PTER					
		160				
4973	PTITE					
		237				
4974	PTPMT					
		239				
3549	PTRET					
		3555				
5154	SEAD					
		375				
120	SELF					
		686	688	995	997	1375
4589	SMAD1					
		4592				
4595	SMAD2					
		4599	4609			
4606	SMAD3					
		4596				
4585	SMAD					
		206				
4602	SMADE					
		4593				
5153	SSAD					
		374				
2965	TO					
		2924	2931	2943		
2966	T1					
		2929	3128			
2967	T2					
		3130				
2968	T3					
		NOT USED				
3208	TAI1					
		273				
3209	TAI2					
		274				
5006	TBUF					
		250				
5008	TCDS					
		252				
3236	TCIW					
		278				
5007	TCIWC					
		251				
3181	TCOPY					
		163				
5021	TCTP					
		265				
3139	TD1					
		3129	3136			
3133	TD2					
		3115	3131			

3137	TD3	3132	
3146	TDCH	3107	
3111	TDIAL	165	
3141	TDSEL	3134	3138
5014	TDWL	258	
3214	TECI	276	
5013	TEMP	257	
5012	TEOC	256	
5204	TERR0	NOT USED	
5205	TERR1	NOT USED	
5200	TERR	266	
5201	TERRF	NOT USED	
2995	TEZ	2992	
5011	TFST	255	
3011	TGZ	2994	3004
3207	THI	272	
3210	TIMX	275	
5009	TIN	253	
5010	TINT	254	
3182	TLOOP	3189	
3006	TLZ	2993	
2924	TMSG	2915	2918
3041	TNCY	164	
3135	TND2	3113	
3160	TNDC	3050	3144
5020	TNTP	264	

2975	TNXT	2932	2998	3002					
3018	TOL	2984	2988						
3291	TPE	281							
2909	TPEX	162							
5017	TPM1	261							
5018	TPM2	262							
5019	TPM3	263							
3252	TPTAB	280							
3108	TPUE	3091	3094						
2960	TPSET	2944	2954						
3206	TSCI	271							
3070	TSD2	3065							
3078	TSD3	3073							
2940	TSST	2925							
3060	TSYNC	NOT USED							
3089	TSYNCE	3068	3076	3081	3084				
3086	TSYNCR	3057	3067	3075	3083				
4114	VC01	4104	4109						
4130	VC02	4116							
4210	VC31	4291							
4211	VC32	4296							
4212	VC41	4315							
4213	VC42	4320							
4984	VCBT	241							
4529	VCBUS	4409	4521						
4392	VCCO	4243	4264	4300	4324	4389	4418	4433	4485	4503

4389	VCCP	4350 4519
4987	VCDB	245
4221	VCDHK	4163
4983	VCDT	244
4171	VCER1	4069
4177	VCER2	4144 4146 4148
4183	VCER3	4151 4153 4155 4157
4189	VCER4	4227 4230 4233
4545	VCER	4411
4164	VCGTF	4140
4199	VCHD	4067
4337	VCHE	NOT USED
4283	VCHFE	4162 4250 4255 4258 4271 4277
4378	VCK0	4384
4332	VCK1	4373
4417	VCK21	4370
4416	VCK2	4381 4385
4430	VCK3	4369 4377 4380
4447	VCK5	4428
4471	VCK6	4477
4475	VCK7	4466
4479	VCK8	4463 4474
4500	VCK9	4462 4470 4473
4354	VCK	4525
4344	VCKFE	4276
4060	VCLAS	158

4200	VCLD									
		4068								
4993	VCLOG									
		243								
4408	VCM15									
		4371	4372	4374	4376	4383	4464	4465	4467	4469	4476
4202	VCMA									
		4142								
4204	VCMOF									
		4159	4161							
4203	VCMT									
		4149								
4518	VCNK									
		4498								
4111	VCRET									
		4079	4176	4182	4188	4194				
4912	VCSDB									
		246								
4206	VCST1									
		4275								
4207	VCST2									
		4254								
4208	VCST3									
		4256	4257							
4986	VCVC									
		242	1361							
4538	VCVNB									
		4410	4481	4497	4517					
4955	XBUF									
		220								
4770	XCCE									
		4765								
4727	XCER									
		4716								
4811	XCFT									
		4810	4855							
4854	XCFTE									
		4855								
4959	XCIWC									
		NOT USED								
4960	XCIWP									
		NOT USED								
5174	XERR0									
		NOT USED								
5175	XERR1									
		NOT USED								
5168	XERR									
		221								
5169	XERRF									
		NOT USED								

```
4757  XFO1
                4741
4783  XFO2
                4767
4737  XFOW
                4714
4855  XIMX
                4740  4766
4872  XLOF1
                 201
4884  XLOF2
                 203
4866  XLON1
                 200
4878  XLON2
                 202
4744  XMCE
                4739
4957  XOCW1
              NOT USED
4958  XOCW2
              NOT USED
4691  XRTC
                 166
4712  XSDC
                 168
4956  XSDN
              NOT USED
4702  XWDT
                 167
```

```
*****************************************************************************
*****************************************************************************

                  *****    ****      ****     ****      ****
                  *   *    *   *       *        *         *
  ****     ****   *****    *****      ***      ***        *     ****     ****
                  *        *   *        *        *        *
                  *        *   *    *   *    *   *    *   *
                  *        *****     ****     ****     ****

*****************************************************************************
*****************************************************************************
-- 3 0 --         DATE  10-10-75    TIME  17.284       ID = G5   CS-F.0
```

```
1 0000                    .TITLE PBD33J.'                    ... P A R B 3 3 J ...'
```

```
 4 0000                    .LIST   I
 5 0000                    .ASECT
 6 0000 FF00 A             .=X'FF00                  ; SET LOCATION COUNTER TO TOP PAGE
 7 FF00          ;
 8 FF00 FFF8 A             .=.+X'F8
 9 FFF2 2104 A             JMP     .+5               ; 'CATCHER IN THE RYE'
10 FFF4 2103 A             JMP     .+4               ; 'CATCHER IN THE RYE'
11 FFF6          ;
12 FFF6 FFCF A             .WORD   X'FFCF            ; CCU PCI START ADDRESS + X'CF
13 FFF8 25FF A             JMP     @.-1
14 FFFC          ;
15 FFFC FFD9 A             .WORD   X'FFD9            ; CCU LCI START ADDRESS + X'C8
16 FFFE 25FF A             JMP     @.-1
17 FFFF          ;
18 FFFF 2000 A             JMP     0                 ; RESTART INITIALIZATION
19 FFFF          ;
20 FFFF 21FF A             JMP     .                 ; LOOP FOR CCU PSEUDO HALT
21 0000                    .END

******      0 ERRORS IN ASSEMBLY   ******
```

TOP PAGE OF MEMORY (EE00 TO FFFF)

E6DC 0000

PROGRAM LISTING OF PBD14C - BUS PREEMPT CONTROL UNIT
PROGRAM FOR WASHINGTON INSTALLATION

APPENDIX D. FIGURE C.8 PROGRAM LISTING OF PBD14C - BPCU PROGRAM FOR WASHINGTON INSTALLATION

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

 XXXXX            XXXX              X              XX            XXXXX
 X    X           X   X            XX             X X            X
 XXXXX            XXXXX             X            X  X            X
 X                X    X            X           XXXXXX           X
 X                X    X            X              X             X
 X                XXXXX           XXX             X              XXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


$$    PB14C ENTERED CS-F.0 AT 13.506 FROM REMOTE  0-08-04


0001  $      SNUMB   PB14C
0002  $      IDENT   D00391-IMP16XREF
0003  $$     USERID  D00391$##
0004  $      OPTION  FORTRAN.NOMAP
0005  $$     SELECT  G02771/NATIONAL/POINTER
0006* $      OPTIONS FORTRAN.NOMAP
0007* $$     SELECT  G02771/NATIONAL/ACCOUNT
0009* $      OBJECT                                          Y09.711032175......00
0010* $$     SELECT  LIBRARY/SURCHK/A3XXX
0011* $      OBJECT                                          G13.092021475A3XXX000
0013* $      EXECUTE
0014* $      LIMITS  C1.4K
0015* $      SYSOUT  P*.ORG
0016* $      OPTIONS FORTRAN.NOMAP
0017* $$     SELECT  G02771/NATIONAL/ASMIMP16
0018* $      OBJECT                                          Y10.768070275BINSCH00
0020* $      OBJECT                                          Y10.769070275STRING00
0022* $      OBJECT                                          Y10.769070275LABEL000
0024* $      OBJECT                                          Y10.770070275LCVAL000
0026* $      OBJECT                                          Y10.770070275I2HEX000
0028* $      OBJECT                                          Y10.771070275DEC2I000
0030* $      OBJECT                                          Y10.771070275UNPACK00
0032* $      OBJECT                                          Y10.772070275SYMBOL00
0034* $      OBJECT                                          Y10.772070275HEX2I000
0036* $      OBJECT                                          Y10.773070275ASC2I000
0038* $      OBJECT                                          Y10.773070275FORM0000
0040* $      OBJECT                                          Y10.773070275NXCHAR00
0042* $      OBJECT                                          Y10.774070275PSECT000
0044* $      OBJECT                                          Y10.774070275REMOVE00
0046* $      OBJECT                                          Y10.775070275OR000000
0048* $      OBJECT                                          Y10.775070275AND00000
0050* $      OBJECT                                          Y10.775070275INSERT00
0052* $      OBJECT                                          Y10.776070275ERRCOL00
0054* $      OBJECT                                          Y10.776070275EVAL0000
0056* $      OBJECT                                          Y10.777070275DATA0000
0058* $      OBJECT                                          Y10.777070275ASSIGN00
0060* $      OBJECT                                          Y10.777070275PRFORM00
0062* $      OBJECT                                          Y10.778070275LCNTL000
0064* $      OBJECT                                          Y10.778070275MEMREF00
0066* $      OBJECT                                          Y10.778070275MISCIN00
0068* $      OBJECT                                          Y10.779070275LCMOD000
0070* $      OBJECT                                          Y10.779070275XMTREC00
0072* $      OBJECT                                          Y10.779070275NXPAGE00
0074* $      OBJECT                                          Y10.780070275NXITEM00
```

```
0076* $     OBJECT                                                  Y10.780070275......00
0078* $     OBJECT                                                  Y10.780070275ACNTL000
0080* $     OBJECT                                                  Y10.781070275LIST0000
0082* $     OBJECT                                                  Y10.781070275REGINT00
0084* $     OBJECT                                                  Y10.782070275INST0000
0086* $     OBJECT                                                  Y10.782070275000000000
0088* $     OBJECT                                                  Y10.782070275SYMVAL00
0090* $     OBJECT                                                  Y10.783070275EXTSYM00
0092* $     OBJECT                                                  Y10.783070275NXTSYM00
0094* $     OBJECT                                                  Y10.783070275ADDSYM00
0096* $     OBJECT                                                  Y10.784070275ADPART00
0098* $     OBJECT                                                  Y10.784070275ERREPT00
0100* $     OBJECT                                                  Y10.784070275DIRECT00
0102* $     OBJECT                                                  Y10.785070275BOCNTL00
0104* $     OBJECT                                                  Y10.785070275GENRLM00
0106  $     EXECUTE
0107  $     SET      6,19
0108  $     LIMITS   15,23K,,7500
0109  $     REMOTE   06
0110  $     FILE     10,X1S,40L
0111  $$    PRMFL    09,R/W,S,D00391/AOUT
0112  $$    SELECT   D00391/EAC1002$######
0113* $     INCODE   IBMEL
0114  $     INCODE   IBMEL
0115  $$    SELECT   D00391/EBP0826$######
0116* $     INCODE   IBMEL
0117  $     INCODE   IBMEL
0118  $$    SELECT   D00391/EER1001$######
0119* $     INCODE   IBMEL
0120  $     INCODE   IBMEL
0121  $$    SELECT   D00391/EES0826$######
0122* $     INCODE   IBMEL
0123  $     INCODE   IBMEL
0124  $$    SELECT   D00391/EPP0819$######
0125* $     INCODE   IBMEL
0126  $     INCODE   IBMEL
0127  $$    SELECT   D00391/ETP1002$######
0128* $     INCODE   IBMEL
0129  $     INCODE   IBMEL
0130  $$    SELECT   D00391/EPR1010$######
0131* $     INCODE   IBMEL
0132  $     INCODE   IBMEL
0133  $$    SELECT   D00391/EVC1001$######
0134* $     INCODE   IBMEL
0135  $     INCODE   IBMEL
0136  $$    SELECT   D00391/EEB0826$######
0137* $     INCODE   IBMEL
0138  $     INCODE   IBMEL
0139  $$    SELECT   D00391/EBX0826$######
0140* $     INCODE   IBMEL
0141  $     INCODE   IBMEL
0142  $$    SELECT   D00391/EDB1010$######
0143* $     INCODE   IBMEL
0144  $     IF       /19,ENDJOB
0145  $     OPTION   FORTRAN,NOMAP
0146  $     LIBRARY  L1
0147  $$    SELECT   G02771/NATIONAL/XREF
0148* $     OBJECT                                                  Y16.706071475......00
0150  $     EXECUTE
0151  $     LIMIT    ,20K
0152  $$    PRMFL    L1,R,R,USERLIB/UTILITY/BUTIL
0153  $     FILE     10,X1R
0154  $     FILE     11,X2R,40L
0155  $     FILE     51,X3R,44R
```

```
0156  $      ENDJOB
 IO    CARD    INT    JO   0054

 * BEGIN ACTIVITY -01-  GELOAD   10/10/75   SW=010000000000
  * NORMAL TERMINATION     AT 011743 I=5020 SW=010000000000

     START 13.517      LINES     21      PROC  0.0001      I/O   0.000    IU  5   MEMORY    5K
     STOP  13.517      LIMIT   5000      LIMIT 0.0100     LIMIT           CU  5   M*T       17
     SWAP   0.000
     LAPSE  0.000   FC D TYPE     BUSY    IP/AT    FP/RT    IS/#C MS/#E    ADDRESS T#

                    R* R D191 *     21        0        0        1     1    0-08-01
                    P*  SYOUT
                    L* R D191 *    273        0        0      528   528R   0-08-05

     LIST      21 LINES AT STA.  G5

RESOURCE UNITS USED, ACTIVITY -01-      0.00
***RESOURCE UNITS USED, JOB TOTAL-      0.00

 * BEGIN ACTIVITY -02-  GELOAD   10/10/75   SW=014000200000
  * NORMAL TERMINATION     AT 013613 I=5020 SW=014000200000

     START 13.531      LINES   5625      PROC  0.0659      I/O   0.006    IU  5   MEMORY   23K
     STOP  13.659      LIMIT   7500      LIMIT 0.1500     LIMIT           CU  5   M*T    11566
     SWAP   0.000
     LAPSE  0.128   FC D TYPE     BUSY    IP/AT    FP/RT    IS/#C MS/#E    ADDRESS T#

                    I* R D191 *   8950        0      228      228   228    0-08-01
                    R* R D191 *   1811        0        0       46    46    0-08-01
                    06  SYOUT
                    10 S D191 *   2436        0        0      480   480    0-08-02
                    09 R D191 P    724        0       16       20    20    0-08-05
                    P*  SYOUT
                    L* R D191 *    997        0        0      528   528R   0-08-05

     LIST      22 LINES AT STA.  G5
     RC-06   5603 LINES AT STA.  G5

RESOURCE UNITS USED, ACTIVITY -02-      2.53
***RESOURCE UNITS USED, JOB TOTAL-      2.54

 * BEGIN ACTIVITY -03-  GELOAD   10/10/75   SW=000000200000
SORT ENGAGED - VERSION: F/8
          FILE 11 UNSUITABLE FOR COLLATION.
MEMORY: 023416  LINKS: 00044 INPUT: 0320/DOUBLE  OUTPUT: 0320/DOUBLE  COLLATION: 0362/DOUBLE  TOURNAMENT: 000541 ENTRIES
          END OF FILE ON 10.  RECORDS READ = 00006146.  RECORDS ACCEPTED = 00006146.  RECORDS DELETED = 00000000.
          MERGING 000006 STRINGS 08 WAYS.
          LINK UTILIZATION.       ALLOCATED 00044   BORROWED 00000   USED 00007
          RECORDS INPUT = 00006146.  RECORDS OUTPUT = 00006146.
SORT TERMINATES.
  * NORMAL TERMINATION     AT 013403 I=5020 SW=000000200000

     START 13.661      LINES   1995      PROC  0.0046      I/O   0.004    IU  5   MEMORY   20K
     STOP  13.677      LIMIT   5000      LIMIT 0.0500     LIMIT           CU  5   M*T     1122
     SWAP   0.001
     LAPSE  0.015   FC D TYPE     BUSY    IP/AT    FP/RT    IS/#C MS/#E    ADDRESS T#

                    10 R D191 *   2189        0        0      480   480    0-08-02
                    R* R D191 *     96        0        0        3     3    0-08-01
                    L1 R D191 P     48        0        0      120   120R   0-08-03
```

```
S1 R 0191 *   3835        0        0      570    520K   0-08-02
D* SYNIT
L* R 0191 *   .044        0        0      [illegible]    [illegible]   0 00 05

LIST      26 LINES AT STA.  G5
RC-06   1969 LINES AT STA.  G5

RESOURCE UNITS USED, ACTIVITY -03-      0.26
***RESOURCE UNITS USED, JOB TOTAL-      2.80
```

SNUMB = PB14C, ACTIVITY # = 01, REPORT CODE = 74, RECORD COUNT = 000021

```
ORIGIN   DATE MODULE ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION

                SUBPROGRAMS INCLUDED IN DECK.

            $     OPTION  FORTRAN,NOMAP
            $     OPTIONS FORTRAN,NOMAP

                SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

                                RANGE              SIZE
            ALLOCATED CORE    000000 THRU 011777   012000
               RELOCATABLE    007062 THRU 011777   002716
            $     SYSOUT  P*,ORG

            FCB AND BUFFER SPACE
               AVAILABLE      000101 THRU 007061   006761
               FILE CTRL BLKS 006732 THRU 007062   000131
               MAXIMUM BUFFER SPACE REQUIRED       001202

                3K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN     730517 F/8
            000204 LOCATIONS REQUIRED FOR LOAD TABLE
            EXECUTION PROGRAM ENTERED AT  011772   THROUGH  .FSETU
```

SNUMB = PB14C, ACTIVITY # = 02, REPORT CODE = 74, RECORD COUNT = 000022

```
PB14C 02  10-10-75   13.531                                                                                    PAGE      1
  ORIGIN   DATE MODULE ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                       SUBPROGRAMS INCLUDED IN DECK.

                 $      OPTIONS FORTRAN,NOMAP

                       SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

                                          RANGE                SIZE
                 ALLOCATED CORE     000000 THRU 055777      056000
                    RELOCATABLE     004422 THRU 055777      051356
                 $      REMOTE  06
                 $      FILE    10,X15,40L
                 $      PRMFL   09,R/W,S,D00391/AOUT

                 FCB AND BUFFER SPACE
                    AVAILABLE       000101 THRU 004421      004321
                    FILE CTRL BLKS  004170 THRU 004422      000233
                    MAXIMUM BUFFER SPACE REQUIRED           003105

                     23K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN     730517 F/8
                 001446 LOCATIONS REQUIRED FOR LOAD TABLE
                 EXECUTION PROGRAM ENTERED AT  033506   THROUGH  .FSETU
```

SNUMB = PB14C, ACTIVITY # = 02, REPORT CODE = 06, RECORD COUNT = 005603

```
1 0000          .TITLE PBD14C,'          ... P B D 1 4 C ...'
```

```
 4 0000                    .LIST  LAC
 5 0000          ;
 6 0000          ;                    ... A S S E M B L Y   C O N S T A N T S ...
 7 0000          ;
 8 0000 0000 A AC0      =       0              ; ACCUMULATOR 0
 9 0000 0001 A AC1      =       1              ; ACCUMULATOR 1
10 0000 0002 A AC2      =       2              ; ACCUMULATOR 2
11 0000 0003 A AC3      =       3              ; ACCUMULATOR 3
12 0000         ;
13 0000 0001 A FC1      =       X'01           ; FUNCTION CODE 1, WRITE WORD 1
14 0000 0002 A FC2      =       X'02           ; FUNCTION CODE 2, WRITE WORD 2
15 0000 0003 A FC3      =       X'03           ; FUNCTION CODE 3, READ WORD 1
16 0000 0004 A FC4      =       X'04           ; FUNCTION CODE 4, READ WORD 2
17 0000         ;
18 0000 0010 A FC10     =       X'10           ; FUNCTION CODE 10, READ THUMBWHEEL 6
19 0000 0011 A FC11     =       X'11           ; FUNCTION CODE 11, READ THUMBWHEEL 1
20 0000 0012 A FC12     =       X'12           ; FUNCTION CODE 12, READ THUMBWHEEL 2
21 0000 0013 A FC13     =       X'13           ; FUNCTION CODE 13, READ THUMBWHEEL 3
22 0000 0014 A FC14     =       X'14           ; FUNCTION CODE 14, READ THUMBWHEEL 4
23 0000 0015 A FC15     =       X'15           ; FUNCTION CODE 15, READ THUMBWHEEL 5
24 0000         ;
25 0000         ;                                   NOTE: FUNCTION CODES ENDING IN
26 0000         ;                                         6 CANNOT BE USED FOR RIN
27 0000         ;                                         INSTRUCTIONS DUE TO
28 0000         ;                                         CONFLICT WITH TTY ROUTINE
29 0000         ;
30 0000 0017 A FC17     =       X'17           ; FUNCTION CODE 17, READ THUMBWHEEL 7
31 0000         ;
32 0000 0041 A FC41     =       X'41           ; FUNCTION CODE 41, RESET PULSE
33 0000 0042 A FC42     =       X'42           ; FUNCTION CODE 42, START PULSE
34 0000 0043 A FC43     =       X'43           ; FUNCTION CODE 43, STOP PULSE
35 0000 0044 A FC44     =       X'44           ; FUNCTION CODE 44, TURN ON LIGHT 1
36 0000 0045 A FC45     =       X'45           ; FUNCTION CODE 45, TURN OFF LIGHT 1
37 0000 0046 A FC46     =       X'46           ; FUNCTION CODE 46, TURN ON LIGHT 2
38 0000 0047 A FC47     =       X'47           ; FUNCTION CODE 47, TURN OFF LIGHT 2
39 0000         ;
40 0000         ;                    ... DISPLACEMENT CONSTANTS ...
41 0000 FFF4 A DM12     =       -12
42 0000 FFF9 A DM7      =       -7
43 0000 FFFC A DM4      =       -4
44 0000 FFFE A DM2      =       -2
45 0000 FFFF A DM1      =       -1
46 0000 0000 A D0       =       0
47 0000 0001 A D1       =       1
48 0000 0002 A D2       =       2
49 0000 0003 A D3       =       3
50 0000 0004 A D4       =       4
51 0000 0005 A D5       =       5
52 0000 0006 A D6       =       6
53 0000 0007 A D7       =       7
54 0000 0008 A D8       =       8
55 0000 0009 A D9       =       9
```

```
 56 0000 000A A D10     =      10
 57 0000 000B A D11     =      11
 58 0000 000C A D12     =      12
 59 0000 000D A D13     =      13
 60 0000 000E A D14     =      14
 61 0000 000F A D15     =      15
 62 0000 0010 A D16     =      16
 63 0000 0011 A D17     =      17
 64 0000 0012 A D18     =      18
 65 0000 0013 A D19     =      19
 66 0000 0014 A D20     =      20
 67 0000 0015 A D21     =      21
 68 0000 0016 A D22     =      22
 69 0000 0017 A D23     =      23
 70 0000 0018 A D24     =      24
 71 0000 0019 A D25     =      25
 72 0000 001A A D26     =      26
 73 0000 0074 A D116    =      116
 74 0000 0076 A D118    =      118
 75 0000          ;
 76 0000 0001 A DETS    =      1              ; NUMBER OF DETECTORS FOR PBD14C .......
 77 0000          ;
 78 0000 0019 A DPMD    =      25             ; MINIMUM DURATION OF A PREEMPT
 79 0000          ;                                ( 0.1 SEC UNITS)
 80 0000          ;
 81 0000          ;
 82 0000          ;                       ... IMMEDIATE DATA CONSTANTS ...
 83 0000 FF92 A CM110   =      -110
 84 0000 FF9C A CM100   =      -100
 85 0000 FFEC A CM20    =      -20
 86 0000 FFF0 A CM16    =      -16
 87 0000 FFF6 A CM10    =      -10
 88 0000 FFFA A CM6     =      -6
 89 0000 FFFE A CM2     =      -2
 90 0000 FFFF A CM1     =      -1
 91 0000 0000 A C0      =      0
 92 0000 0001 A C1      =      1
 93 0000 0002 A C2      =      2
 94 0000 0003 A C3      =      3
 95 0000 0004 A C4      =      4
 96 0000 0005 A C5      =      5
 97 0000 0006 A C6      =      6
 98 0000 0007 A C7      =      7
 99 0000 0008 A C8      =      8
100 0000 000F A C15     =      15
101 0000 0030 A C48     =      48
102 0000 007D A C125    =      125
```

```
                    .PAGE
          ;
          ;                    ... BOC CONDITION CODES ...
0000 0001 A CCEZ    =       X'1          ; CONDITION CODE 1 (AC0).EQ.0
0000 0002 A CCZP    =       X'2          ; CONDITION CODE 2 (AC0) ZERO, POSITIVE
0000 0003 A CCB0    =       X'3          ; CONDITION CODE 3 AC0 BIT 0 = 1
0000 0004 A CCB1    =       X'4          ; CONDITION CODE 4 AC0 BIT 1 = 1
0000 0005 A CCNZ    =       X'5          ; CONDITION CODE 5 (AC0).NE.0
0000 0008 A CCSF    =       X'8          ; CONDITION CODE 8 STACK FULL
0000 000B A CCZN    =       X'B          ; CONDITION CODE 11 (AC0).LE.0
0000 000C A CCCU    =       X'C          ; CONDITION CODE 12 CCU IN SYSTEM
0000 000D A CCPH    =       X'D          ; CONDITION CODE 13 INITIALIZATION
0000                                     ;    PSEUDO HALT
0000 000E A CCCF    =       X'E          ; CONDITION CODE 14 CATCH ON THE FLY
          ;
          ;                    ... CONTROL FLAGS ...
0000 0001 A INEN    =       1            ; INTERRUPT ENABLE FLAG
0000 0002 A SELF    =       2            ; SELECT FLAG
          ;
```

```
 122 0000                .PAGE
 123 0000        ;                   ... BASE PAGE CONSTANTS ...
 124 0000        ;                       FOR UNUSED PROGRAM ENTRY POINTS IN PBD14C
 125 0000 1A88 A PTER    =       X'1A88
 126 0000 1AE6 A PREQ    =       X'1AE6
 127 0000 1A4E A PCIM    =       X'1A4E
 128 0000        ;
```

```
131 0000                .LIST   LBP
132 0000        ;
133 0000        ;                   ... B A S E   P A G E ...
134 0000                .BSECT
135 0000        ;                   ... TRANSFER TO RESTART INITIALIZATION ROUTINE...
136 0000 2420 A         JMP     @AERSI
137 0001        ;
138 0001        ;                   ... TRANSFER TO GENERAL INTERRUPT SERVICING ROUTINE
139 0001 2421 A         JMP     @AEGIS
140 0002        ;
```

```
0002                    .LIST   LBP
0002            ;
0002                    .ASECT
0000 0020 A             .=X'20
0020 1000 A AERSI:  .WORD   ERSI        ; RESTART INITIALIZATION ROUTINE
0021 1041 A AEGIS:  .WORD   EGIS        ; GENERAL INTERRUPT SERVICING ROUTINE
0022 10C2 A AECCU:  .WORD   ECCU        ; CCU INITIALIZATION ROUTINE
0023 10E0 A AEMAD:  .WORD   EMAD        ; INITIAL RAM ADDRESS TEST ROUTINE
0024 1E00 A AEBKD:  .WORD   EBKD        ; BACKGROUND PROGRAM MONITOR
0025 10BF A AECER:  .WORD   ECER        ; CLOCK ERROR ROUTINE
0026 10F3 A AESCH:  .WORD   ESCH        ; EXECUTIVE SCHEDULING ROUTINE
0027 1200 A AEDIS:  .WORD   EDIS        ; DISCONNECT MONITOR
0028 1212 A AECOM:  .WORD   ECOM        ; COMMAND MONITORING ROUTINE
0029 123F A AESYN:  .WORD   ESYN        ; SYNCHRONIZATION ROUTINE
002A 1282 A AEMIC:  .WORD   EMIC        ; INTERCONNECT DATA MONITOR ROUTINE
002B 12A3 A AETIM:  .WORD   ETIM        ; EXECUTIVE TIMING ROUTINE
002C 131B A AEEOC:  .WORD   EEOC        ; END OF CYCLE ROUTINE
002D 12E0 A AESDW:  .WORD   ESDW        ; STATUS DISPLAY ROUTINE
002E 12B2 A AEDET:  .WORD   EDET        ; DETECTOR MONITORING ROUTINE
002F 110F A AEADC:  .WORD   EADC        ; AUTODISCONNECT ROUTINE
0030 1A00 A APMON:  .WORD   PMON        ; PREEMPT MONITOR ROUTINE
0031 1C19 A AVCLAS: .WORD   VCLAS       ; VEHICLE CLASSIFICATION ROUTINE
0032 1AE6 A APREQ:  .WORD   PREQ        ; PREEMPT REQUEST ROUTINE
0033 1A88 A APTER:  .WORD   PTER        ; PREEMPT TERMINATION ROUTINE
0034 1A4F A APCIM:  .WORD   PCIM        ; ROUTINE USED TO RESTORE CI
0035 1800 A ATPEX:  .WORD   TPEX        ; TIMING PLAN EXECUTION ROUTINE
0036 18C9 A ATCOPY: .WORD   TCOPY       ; TIMING PLAN COPY ROUTINE
0037 1866 A ATNCY:  .WORD   TNCY        ; NEXT TIMING PLAN CYCLE ROUTINE
0038 1896 A ATDIAL: .WORD   TDIAL       ; INIT ENTRY TO TNCY ROUTINE
0039 1F28 A AXRTC:  .WORD   XRTC        ; RTC DATA OUTPUT SEQUENCE ROUTINE
003A 1F32 A AXWDT:  .WORD   XWDT        ; RESET WATCHDOG TIMER ROUTINE
003B 1F36 A AXSDC:  .WORD   XSDC        ; FORMAT OUTPUT COMMANDS ROUTINE
003C 1F00 A ABRTC:  .WORD   BRTC        ; RTC DATA INPUT SEQUENCE ROUTINE
003D 1143 A ACOTF:  .WORD   COTF        ; CATCH ON THE FLY ROUTINE FOR CHECKOUT ........
003E            ;
003E            ;                   ... PROG. PANEL PROGRAM POINTERS ...
003E 1F0D A ABPAN:  .WORD   BPAN        ; PROG. PANEL PARAMETER INPUT ROUTINE
003F 1400 A AEPAN:  .WORD   EPAN        ; PROG. PANEL INITIALIZATION ROUTINE
0040 146E A AEPPM:  .WORD   EPPM        ; PROG PANEL MONITOR ROUTINE
0041 14C4 A AEPP1:  .WORD   EPP1        ; ROUTINE TO PROCESS PROG. PANEL PLAN 1
0042 1578 A AEPP2:  .WORD   EPP2        ; ROUTINE TO PROCESS PROG. PANEL PLAN 2
0043 162C A AEPP3:  .WORD   EPP3        ; ROUTINE TO PROCESS PROG. PANEL PLAN 3
0044 16E0 A AEPUP:  .WORD   EPUP        ; UPDATE PROG. PANEL DATA
0045 1733 A AEPBI2: .WORD   EPBI2       ; CONVERT 2 BINARY-CODED DIGITS
0046 1726 A AEPBI3: .WORD   EPBI3       ; CONVERT 3 BINARY-CODED DIGITS
0047            ;
0047 174D A AEM10:  .WORD   EM10        ; ROUTINE TO MULTIPLY AN INTEGER BY 10
0048 1752 A AEM24:  .WORD   EM24        ; ROUTINE TO MULTIPLY AN INTEGER BY 24
0049            ;
0049 1340 A AERE0:  .WORD   ERE0        ; ERROR RECORDING ROUTINE
004A 1349 A AERE1:  .WORD   ERE1        ; ERROR RECORDING ROUTINE
004B 1352 A AERE2:  .WORD   ERE2        ; ERROR RECORDING ROUTINE
```

```
004C 135B A AERE3:  .WORD   ERE3      ; ERROR RECORDING ROUTINE
004D 1364 A AERE4:  .WORD   ERE4      ; ERROR RECORDING ROUTINE
004E 136D A AERE5:  .WORD   ERE5      ; ERROR RECORDING ROUTINE
004F 1376 A AERE6:  .WORD   ERE6      ; ERROR RECORDING ROUTINE
0050 137F A AERE7:  .WORD   ERE7      ; ERROR RECORDING ROUTINE
0051 1388 A AERE8:  .WORD   ERE8      ; ERROR RECORDING ROUTINE
0052 1391 A AERE9:  .WORD   ERE9      ; ERROR RECORDING ROUTINE
0053 139A A AERE10: .WORD   ERE10     ; ERROR RECORDING ROUTINE
0054 13A3 A AERE11: .WORD   ERE11     ; ERROR RECORDING ROUTINE
0055 13AC A AERE12: .WORD   ERE12     ; ERROR RECORDING ROUTINE
0056        ;
0056 1F97 A AXLON1: .WORD   XLON1     ; ROUTINE TO TURN STATUS LIGHT ON
0057 1F9A A AXLOF1: .WORD   XLOF1     ; ROUTINE TO TURN STATUS LIGHT OFF
0058 1F9D A AXLON2: .WORD   XLON2     ; ROUTINE TO TURN STATUS LIGHT ON
0059 1FA0 A AXLOF2: .WORD   XLOF2     ; ROUTINE TO TURN STATUS LIGHT OFF
005A        ;
005A        ;               ... SELF TEST ENTRY POINTS ...
005A 1E0C A ASMAD:  .WORD   SMAD      ; RAM ADDRESS TEST
```

```
005B                    .LIST   LBP
005B 0068 A             .=X'68
0068            ;
0068            ; DATA INPUT CPC DATA BASE POINTERS
0068            ;
0068 3100 A ABBUF:  .WORD   BBUF            ; DATA INPUT CPC DATA BASE
0069 3100 A ABDIN:  .WORD   BDIN            ; DISCRETE INPUT WORD
006A 32BE A ABERR:  .WORD   BERR            ; ERROR BUFFER
006B            ;
006B            ; DATA OUTPUT CPC DATA BASE POINTERS
006B            ;
006B 3119 A AXBUF:  .WORD   XBUF            ; DATA OUTPUT CPC DATA BASE
006C 32BF A AXERR:  .WORD   XERR            ; ERROR BUFFER
006D            ;
006D            ; PREEMPT CPC DATA BASE POINTERS
006D            ;
006D 311E A APBUF:  .WORD   PBUF            ; PREEMPT CPC DATA BASE
006E 311E A APIP:   .WORD   PIP                   ; PREEMPT IN PROGRESS FLAG
006F 311F A APTEM:  .WORD   PTEM                  ; TEMPORARY STORAGE LOCATION
0070 3120 A APTCP:  .WORD   PTCP            ; ADDRESS OF PRIMARY CONTRACTIBLE INT
0071 3121 A APTCS:  .WORD   PTCS            ; ADDRESS OF SECONDARY CONTRACTIBLE INT
0072 3122 A APTEI:  .WORD   PTEI            ; ADDRESS OF EXTENDIBLE INT
0073 3123 A APMIF:  .WORD   PMIF                  ; PREEMPT MONITOR INDEX FLAG
0074 3126 A APFREQ: .WORD   PFREQ                 ; PREEMPT REQUEST FLAG
0075 3127 A APMDT:  .WORD   PMDT                  ; PREEMPT MINIMUM DURATION TIMER
0076 312B A APEMAX: .WORD   PEMAX           ; MAXIMUM EXTENSION FOR PREEMPT
0077 32C4 A APERR:  .WORD   PERR            ; PREEMPT ERROR BUFFER
0078 3128 A APTAVT: .WORD   PTAVT           ; UNEXPIRED PREEMPT EXTENSION TIME
0079 3129 A APTITE: .WORD   PTITE           ; DISP OF INT TO RECEIVE UNEXPIRED
007A                                        ;    PREEMPT EXTENSION TIME
007A 312A A APTPMT: .WORD   PTPMT           ; BASE ADDRESS OF MASTER TIMING PLAN
007B            ;
007B 3130 A AVCBT:  .WORD   VCBT            ; TIMER FOR RECENT BUS CLASS.
007C 3131 A AVCVC:  .WORD   VCVC            ; VEHICLE COUNTERS (ONE PER DETECTOR)
007D 3165 A AVCLOG: .WORD   VCLOG                 ; BEGINNING LOCATION OF VEHICLE LOG
007E 312F A AVCDT:  .WORD   VCDT                  ; LOCATION OF DETECTOR INDEX WORD
007F 3151 A AVCDB:  .WORD   VCDB            ; TEMPORARY DATA BASE
0080 3000 A AVCSDB: .WORD   VCSDB           ; BUS DETECTOR SIGNATURE DATA BASE
0081            ;
0081            ;TIMING PLAN CPC DATA BASE POINTERS
0081            ;
0081 3171 A ATBUF:  .WORD   TBUF                  ; BEGINNING LOC OF TIMING PLAN DATA BASE
0082 3171 A ATCIWC: .WORD   TCIWC           ; CURRENT COMMAND INDEX WORD
0083 3172 A ATCDS:  .WORD   TCDS                  ; CURRENT DIAL SETTING--1, 2, OR 3
0084 3173 A ATIN:   .WORD   TIN                   ;CURRENT INTERVAL INDEX
0085 3174 A ATINT:  .WORD   TINT                  ; CURRENT INTERVAL TIMER
0086 3175 A ATFST:  .WORD   TFST                  ; WAIT FOR SYNC START FLAG
0087 3176 A ATEOC:  .WORD   TEOC                  ; END OF CYCLE FLAG
0088 3177 A ATEMP:  .WORD   TEMP                  ; TEMPORARY LOCATION
0089 3178 A ATDWL:  .WORD   TDWL                  ; MAXIMUM WAIT FOR SYNC START DWELL
008A 008A A ATPP:   .WORD   ATPP            ; POINTER TO MASTER TP POINTERS
008B                                        ;    (MUST PRECEDE ATPM1,ATPM2,ATPM3)
```

```
008B 3178 A ATPM1:  .WORD   TPM1          ; BEGINNING LOC OF MASTER TP, DIAL 1
008C 318F A ATPM2:  .WORD   TPM2          ; BEGINNING LOC OF MASTER TP, DIAL 2
008D 31A3 A ATPM3:  .WORD   TPM3          ; BEGINNING LOC OF MASTER TP, DIAL 3
008E 31B7 A ATNTP:  .WORD   TNTP          ; BEGINNING LOC OF NEXT TP TABLE
008F 31CB A ATCTP:  .WORD   TCTP          ; BEGINNING LOC OF CURRENT TP TABLE
0090 32D9 A ATERR:  .WORD   TERR          ; TIMING PLAN CPC ERROR FLAGS, COUNTERS,
0091                                      ; DATA
0091        ;
0091        ; TIMING PLAN TABLES POINTERS
0091        ;
0091 18D3 A ATSCI:  .WORD   TSCI          ; SYNC.CONTRACTIBLE INTERVAL
0092 18D4 A ATHI:   .WORD   THI           ; TRANSFER/HOLD INTERVAL
0093 18D5 A ATAI1:  .WORD   TAI1        ; FIRST ADJ INT INDEX
0094 18D6 A ATAI2:  .WORD   TAI2        ; SECOND ADJ INT INDEX
0095 18D7 A ATIMX:  .WORD   TIMX          ; MAXIMUM VALUE FOR TIN
0096 18D8 A ATECI:  .WORD   TECI          ; BEGINNING LOC OF EXTENDIBLE/
0097                                      ; CONTRACTIBLE INTERVAL DATA TABLES
0097 1918 A ATCIW:  .WORD   TCIW          ; BEGINNING LOC OF COMMAND INDEX WORD
0098                                      ; TABLE
0098 192C A ATPTAB: .WORD   TPTAB         ; BEGINNING LOC OF TIMING PLAN TABLES
0099 1967 A ATPE:   .WORD   TPE           ; END LOCATION OF TIMING PLAN TABLE ROM
009A        ;
009A        ;EXECUTIVE CPC DATA BASE POINTERS
009A        ;
009A 31DF A AEBUF:  .WORD   EBUF          ; BEGINNING LOC OF EXECUTIVE CPC DATA
009B                                      ; BASE
009B 31E1 A AETCY:  .WORD   ETCY          ; TIMING PLAN CYCLE TIMER
009C 31E0 A AETST:  .WORD   ETST          ; WAIT FOR SYNC START DWELL TIMER
009D 31E4 A AEPST:  .WORD   EPST          ; START PULSE INDICATOR
009E 31E5 A AEPDT:  .WORD   EPDT          ; START PULSE DURATION TIMER
009F 32DE A AEERR:  .WORD   EERR        ; ERROR BUFFER
00A0        ;
00A0 31E6 A AECMF:  .WORD   ECMF        ; TEMPORARY FLAG FOR ECOM
00A1        ;
00A1 31E7 A AESYD:  .WORD   ESYD        ; SYNCH DATA BUFFER
00A2        ;
00A2 31F9 A AER123: .WORD   ER123       ; RESET SELECT INTERNAL VALUE
00A3 31FA A AERNO:  .WORD   ERNO        ; RESET SELECT DATA BUFFER
00A4        ;
00A4 31FD A AED23:  .WORD   ED23        ; DIAL SELECT INTERNAL VALUE
00A5 31FE A AEDNO:  .WORD   EDNO        ; DIAL SELECT DATA BUFFER
00A6 3201 A AEDMIT: .WORD   EDMIT       ; DETECTOR MONITORING INTERVAL TIMER
00A7 3202 A AEDETT: .WORD   EDETT       ; TEMP STORAGE FOR DET MONITORING
00A8 333C A AEDERR: .WORD   EDERR       ; DETECTOR ERROR BUFFER
00A9 3C00 A AERAM:  .WORD   X'3C00      ; FIRST LOCATION IN FREE RAM (3C00-3FFF)
```

```
00AA                .LIST   LBP
00AA        ;
00AA 3203 A AESAV:  .WORD   ESAV        ; GENERAL INTERRUPT SAVE AREA POINTERS
00AB 3205 A AESV2:  .WORD   ESV2
00AC 3207 A AESVF:  .WORD   ESVF
00AD 3218 A AESVS:  .WORD   ESVS
00AE        ;
00AE 3219 A ACSAV:  .WORD   CSAV        ; SNAPSHOT SAVE AREA
00AF 321B A ACSV2:  .WORD   CSV2
00B0 321D A ACSVF:  .WORD   CSVF
00B1 322E A ACSVS:  .WORD   CSVS
00B2        ;
00B2 3231 A AISV2:  .WORD   ISV2        ; STACK FULL SAVE AREA POINTERS
00B3 3233 A AISTK1: .WORD   ISTK1
00B4 3234 A AISTK2: .WORD   ISTK2
00B5 3235 A AISTK3: .WORD   ISTK3
00B6 3236 A AISTK4: .WORD   ISTK4
00B7        ;
00B7 3245 A AEDSV2: .WORD   EDSV2       ; DISCONNECT SAVE AREA POINTERS
00B8 3247 A AEDSVF: .WORD   EDSVF
00B9 3258 A AEDSVS: .WORD   EDSVS
00BA        ;
00BA 3259 A ACTST:  .WORD   CTST        ; OPTIONAL USE SAVE AREA POINTER
```

```
00BB                    .LIST   LBP
00BB          ;
00BB          ;                    ... PROG. PANEL DATA POINTERS ...
00BB 3103 A ABPPD:  .WORD   BPPD                    ; PANEL DISCRETE INPUT WORD
00BC 3104 A ABPD1:  .WORD   BPD1                    ; TW SWITCH DATA INPUT     DIAL 1
00BD 310B A ABPD2:  .WORD   BPD2                    ; TW SWITCH DATA INPUT     DIAL 2
00BE 3112 A ABPD3:  .WORD   BPD3                    ; TW SWITCH DATA INPUT     DIAL 3
00BF          ;
00BF 3261 A AEPD1:  .WORD   EPD1                    ; PANEL WORKING BUFFER,    DIAL 1
00C0 3268 A AEPD2:  .WORD   EPD2                    ; PANEL WORKING BUFFER,    DIAL 2
00C1 326F A AEPD3:  .WORD   EPD3                    ; PANEL WORKING BUFFER,    DIAL 3
00C2          ;
00C2 3276 A AEPCD1: .WORD   EPCD1                   ; PANEL UPDATE BUFFER,     DIAL 1
00C3 3280 A AEPCD2: .WORD   EPCD2                   ; PANEL UPDATE BUFFER,     DIAL 2
00C4 328A A AEPCD3: .WORD   EPCD3                   ; PANEL UPDATE BUFFER,     DIAL 3
00C5          ;
00C5 3294 A AEPFT:  .WORD   EPFT                    ; FIXED INTERVAL SUMS
00C6          ;
00C6 329B A AEPUF:  .WORD   EPUF                    ; PANEL UPDATE FLAG
00C7 329C A AEPCF:  .WORD   EPCF            ; PROG PANEL MONITOR COMPUTATION FLAG
00C8 329D A AEPIR:  .WORD   EPIR            ; STATE OF PROG PANEL INPUT REQ SWITCH
00C9          ;
00C9          ;                    ... DATA CONSTANTS ...
00C9 146A A AEPCY:  .WORD   EPCY            ; MAXIMUM CYCLE LENGTH
00CA 146B A AEPIN1: .WORD   EPIN1           ; LIMIT ON MAXIMUM EXTENSION
00CB 146D A AE100:  .WORD   E100            ; PERCENTAGE CONVERSION CONSTANT
00CC 146C A AE10:   .WORD   E10             ; CONSTANT FOR DIVISION
00CD          ;
00CD 3297 A AEP1E:  .WORD   EP1E            ; ERROR FLAG FOR PLAN 1
00CE 3298 A AEP2E:  .WORD   EP2E            ; ERROR FLAG FOR PLAN 2
00CF 3299 A AEP3E:  .WORD   EP3E            ; ERROR FLAG FOR PLAN 3
00D0 329A A AEPDVE: .WORD   EPDVE           ; ERROR FLAG FOR DATA VALIDITY
00D1          ;
00D1 32EB A AEPEF1: .WORD   EPEF1           ; PROG. PANEL ERROR DATA FOR PLAN 1
00D2 3306 A AEPEF2: .WORD   EPEF2           ; PROG. PANEL ERROR DATA FOR PLAN 2
00D3 3321 A AEPEF3: .WORD   EPEF3           ; PROG. PANEL ERROR DATA FOR PLAN 3
```

```
00D4                .LIST   LBP
00D4        ;
00D4        ;                   ... RAM ADDRESS TEST DATA POINTERS ...
00D4 329E A ASSAD:  .WORD   SSAD            ; RAM ADDRESS TEST START ADDRESS
00D5 329F A ASEAD:  .WORD   SEAD            ; RAM ADDRESS TEST END ADDRESS
00D6        ;
00D6        ; END OF DATA BASE POINTER
00D6        ;
00D6 337C A ADEND:  .WORD   DEND                    ; END LOCATION OF DATA BASE
```

```
00D7                .LIST   LBP
00D7 00E0 A         .=X'E0
00E0        ;
00E0        ; PERIPHERAL DEVICE ADDRESSES
00E0        ;
00E0 0080 A ALDU:   .WORD   1*128          ; LDU DEVICE ADDRESS
00E1 0100 A ABLSMU: .WORD   2*128          ; BLSMU DEVICE ADDRESS
00E2 0180 A ACMU:   .WORD   3*128          ; CMU DEVICE ADDRESS
00E3 0200 A ATCP:   .WORD   4*128          ; TCP DEVICE ADDRESS
00E4 0280 A ASDU:   .WORD   5*128          ; SDU DEVICE ADDRESS
00E5 0300 A ADMU:   .WORD   6*128          ; DMU DEVICE ADDRESS
00E6 0380 A ASDB:   .WORD   7*128          ; SDB DEVICE ADDRESS
00E7 0400 A ARTC:   .WORD   8*128          ; RTC DEVICE ADDRESS
00E8        ;
00E8 1F80 A ACCU:   .WORD   63*128         ; CCU DEVICE ADDRESS
00E9        ;
00E9        ;                   ... PERIPHERAL DEVICE ADDRESSES FOR PROG. PANEL ...
00E9 0500 A APPDU:  .WORD   10*128     ; DISCRETES
00EA 0580 A APP1U:  .WORD   11*128     ; PLAN 1 PARAMETERS
00EB 0600 A APP2U:  .WORD   12*128     ; PLAN 2 PARAMETERS
00EC 0680 A APP3U:  .WORD   13*128     ; PLAN 3 PARAMETERS
```

```
411 00FD                    .LIST   LBP
412 00ED 00F0 A             .=X'F0
413 00F0          ;
414 00F0          ;                     ... COMMON HEXADECIMAL CONSTANTS ...
415 00F0 7FFF A H7FFF:  .WORD   X'7FFF          ; HEX CONSTANT (BIT 15 RESET)
416 00F1 3000 A H3000:  .WORD   X'3000          ; DATA BASE START ADDRESS
417 00F2 3FFF A H3FFF:  .WORD   X'3FFF          ; BITS 15, 14 RESET
418 00F3                                        ; RAM END ADDRESS
419 00F3 EC00 A HEC00:  .WORD   X'EC00          ; START ADDRESS OF UNUSED CCU RAM
420 00F4 ECCF A HECCF:  .WORD   X'ECCF          ; END ADDRESS OF UNUSED CCU RAM
421 00F5 0200 A H0200:  .WORD   X'0200          ; RTS DO INSTRUCTION
422 00F6 3BFF A H3BFF:  .WORD   X'3BFF          ; END ADDRESS FOR RTC RAM ADDRESS TEST
```

```
423 00F7                .PAGE
424 00F7                .ASECT
425 00F7 1000 A         .=X'1000
426 1000          ;
```

```
1000                    .LIST   LER
1000            ;
1000            ;                   ... E X E C U T I V E   C P C ...
1000            ;
1000            ;                   ... RESTART INITIALIZATION ROUTINE ...
1000            ERSI:
1000 1860 A             BOC     CCSF,ESTK       ; TEST FOR STACKFULL LINE = 1
1001 8CE7 A             LD      AC3,ARTC
1002 4C00 A             LI      AC0,C0
1003 0643 A             ROUT    FC43            ; ASSURE RTC STOPPED
1004 0641 A             ROUT    FC41            ; ASSURE RTC INTERRUPT PULSE RESET
1005            ;
1005 0980 A             PFLG    INEN            ; DISABLE INTERRUPTS
1006            ;
1006            ;                   ... INITIALIZE CCU SERVICING ROUTINE ...
1006 2C22 A             JSR     @AECCU
1007            ;
1007            ;                   ... INITIAL RAM ADDRESS TEST ...
1007 0714 A             CLRST   D4              ; CLEAR MEMORY ADDRESS TEST STATUS FLAG
1008 2C23 A             JSR     @AEMAD
1009            ;
1009 0715 A             CLRST   D5
100A 0716 A             CLRST   D6
100B 0717 A             CLRST   D7
100C            ;
100C 80F5 A             LD      AC0,H0200       ; RTS DO INSTRUCTION
100D B0A9 A             ST      AC0,@AERAM      ; STORE IN FIRST LOCATION OF FREE RAM
100E                                            ;    (ENTRY POINT VIA COTF)
100E            ;
100E            ;                   ... RESET CONFLICT MONITOR ...
100E 8CE2 A             LD      AC3,ACMU
100F 4C00 A             LI      AC0,C0
1010 0641 A             ROUT    FC41
1011            ;
1011            ;                   ... INITIALIZE CPU STACK ...
1011 4EF0 A             LI      AC2,CM16
1012 4700 A             PULL    AC3             ; PULL 16 WORDS FROM STACK
1013 4A01 A             AISZ    AC2,C1
1014 21FD A             JMP     .-2
1015            ;
1015            ;                   ... INITIALIZE STACK FOR BACKGROUND ROUTINES ...
1015 4FFF A             LI      AC3,CM1
1016 4300 A             PUSH    AC3
1017 4300 A             PUSH    AC3             ; PUSH TWO PROTECTION WORDS OF ALL ONES
1018            ;
1018 889A A             LD      AC2,AEBUF
1019 A200 A             ST      AC0,D0(AC2)     ; ZERO RTC COMPUTE FLAG
101A            ;
101A            ;                               ZERO DATA BASE FROM VCSDB TO DEND
101A 4C00 A             LI      AC0,C0
101B 8880 A             LD      AC2,AVCSDB
101C A200 A EZDB:       ST      AC0,(AC2)
```

```
481 101D 4A01 A          AISZ    AC2,C1
482 101E 2100 A          JMP     .+1
483 101F E8D6 A          SKG     AC2,ADEND
484 1020 21FB A          JMP     EZDB
485 1021         ;
486 1021         ;                                  COPY TIMING PLAN TABLES FROM ROM TO READ/
487 1021         ;                                  WRITE DATA BASE
488 1021 8898 A          LD      AC2,ATPTAB
489 1022 8C8B A          LD      AC3,ATPM1
490 1023 8200 A ETCPY:   LD      AC0,(AC2)
491 1024 A300 A          ST      AC0,(AC3)
492 1025 4A01 A          AISZ    AC2,C1
493 1026 4B01 A          AISZ    AC3,C1
494 1027 E899 A          SKG     AC2,ATPE
495 1028 21FA A          JMP     ETCPY
496 1029         ;
497 1029 8898 A          LD      AC2,AETCY          ; INITIALIZE TIMING PLAN CYCLE TIMER
498 102A 8115 A          LD      AC0,HOFFF          ;    MINIMUM TO A LARGE VALUE
499 102B A202 A          ST      AC0,D2(AC2)        ;
500 102C         ;
501 102C         ;                    ... PROGRAMMABLE PANEL ROUTINE ...
502 102C 2C3F A          JSR     @AEPAN
503 102D         ;
504 102D         ;                    ... READ DISCRETE INPUT WORDS, ETC. ...
505 102D 2C3C A          JSR     @ABRTC
506 102E         ;
507 102E         ;                                  DETERMINE INITIAL DIAL SETTING AND COPY
508 102E         ;                                  APPROPRIATE MASTER TIMING PLAN INTO CURRENT
509 102E         ;                                  AND NEXT TIMING PLANS
510 102E 2C38 A          JSR     @ATDIAL
511 102F         ;
512 102F         ;                                  SET INTERVAL COUNTER TO TH INTERVAL
513 102F 9892 A          LD      AC2,@ATHI
514 1030 B884 A          ST      AC2,@ATIN
515 1031         ;
516 1031         ;                                  SET CURRENT INTERVAL TIMER TO LENGTH OF
517 1031         ;                                  CURRENT TH INTERVAL
518 1031 C88F A          ADD     AC2,ATCTP
519 1032 8600 A          LD      AC1,(AC2)
520 1033 B485 A          ST      AC1,@ATINT
521 1034         ;
522 1034         ;                                  COPY APPROPRIATE COMMAND INDEX WORD INTO
523 1034         ;                                  BUFFER(TCIWC)
524 1034 9892 A          LD      AC2,@ATHI
525 1035 C897 A          ADD     AC2,ATCIW
526 1036 8600 A          LD      AC1,(AC2)
527 1037 B482 A          ST      AC1,@ATCIWC
528 1038         ;
529 1038         ;                                  FORMAT OUTPUT WORD
530 1038 2C3B A          JSR     @AXSDC
531 1039         ;
532 1039         ;
```

```
533 1039          ;                    ... INITIALIZATION PSEUDO HALT ...
534 1039 1DFF A EHLT:   BOC     CCPH,EHLT        ; PSEUDO HALT LOOP
535 103A          ;
536 103A          ;
537 103A 4C01 A         LI      AC0,C1
538 103B 8CE7 A         LD      AC3,ARTC
539 103C 0642 A         ROUT    FC42             ; START RTC
540 103D          ;
541 103D          ;                    ... INITIATE BACKGROUND PROGRAMS ...
542 103D 8C24 A         LD      AC3,AEBKD
543 103E 4300 A         PUSH    AC3
544 103F          ;
545 103F 0100 A         RTI     D0               ; RETURN AND ENABLE INTERRUPTS ...
546 1040          ;
547 1040          ;                    ... LOCAL DATA ...
548 1040          ;
549 1040 0FFF A HOFFF:  .WORD   X'0FFF
550 1041          ;
```

```
1041                .LIST   LER
1041        ;
1041        ;                   ... GENERAL INTERRUPT SERVICING ROUTINE ...
1041        EGIS:
1041 181F A         BOC     CCSF,ESTK       ; TEST FOR STACK FULL LINE = 1
1042        ;
1042 B8AB A         ST      AC2,@AESV2      ; SAVE ACCUMULATORS
1043 88AB A         LD      AC2,AESV2
1044 A2FE A         ST      AC0,DM2(AC2)
1045 A6FF A         ST      AC1,DM1(AC2)
1046 AE01 A         ST      AC3,D1(AC2)
1047        ;
1047 4400 A         PULL    AC0             ; SAVE RALU FLAGS
1048 0080 A         PUSHF
1049 5400 A         XCHRS   AC0
104A B0AC A         ST      AC0,@AESVF
104B        ;
104B 4D04 A         LI      AC1,C4          ; SAVE CPU STACK
104C 88AC A         LD      AC2,AESVF       ; STACK FOLLOWS RALU FLAG WORD
104D 4700 A EGIS1:  PULL    AC3             ; PULL STACK FROM TOP TO BOTTOM
104E AE01 A         ST      AC3,D1(AC2)
104F 4700 A         PULL    AC3
1050 AE02 A         ST      AC3,D2(AC2)
1051 4700 A         PULL    AC3
1052 AE03 A         ST      AC3,D3(AC2)
1053 4700 A         PULL    AC3
1054 AE04 A         ST      AC3,D4(AC2)
1055 4A04 A         AISZ    AC2,D4          ; INCREMENT SAVE AREA ADDRESS BY 4
1056 49FF A         AISZ    AC1,CM1         ; DECREMENT PASS COUNTER, SKIP IF DONE
1057 21F5 A         JMP     EGIS1
1058        ;
1058        ;
1058 4C01 A         LI      AC0,C1          ; SAVE SELECT FLAG STATE
1059 58EF A         ROR     AC0,D17
105A 1303 A         BOC     CCB0,EGIS2      ; BRANCH IF SELECT FLAG SET
105B 4C00 A         LI      AC0,C0
105C B0AD A         ST      AC0,@AESVS      ; SELECT FLAG = 0
105D 212D A         JMP     EGIS3
105E 4C01 A EGIS2:  LI      AC0,C1
105F B0AD A         ST      AC0,@AESVS      ; SELECT FLAG = 1
1060 212A A         JMP     EGIS3
1061        ;
1061        ;
1061        ;
1061        ;                   ... CLEAR STACK FULL CONDITION ...
1061        ESTK:
1061 5400 A         XCHRS   AC0
1062 B0B3 A         ST      AC0,@AISTK1     ; SAVE STACK 1
1063 4400 A         PULL    AC0             ; RESTORE AC0
1064 5400 A         XCHRS   AC0
1065 B0B4 A         ST      AC0,@AISTK2     ; SAVE STACK 2
1066 4400 A         PULL    AC0             ; RESTORE AC0
```

```
1067 5400 A          XCHRS   AC0
1068 B0B5 A          ST      AC0,@AISTK3      ; SAVE STACK 3
1069 4400 A          PULL    AC0              ; RESTORE AC0
106A 5400 A          XCHRS   AC0
106B B0B6 A          ST      AC0,@AISTK4      ; SAVE STACK 4
106C 4400 A          PULL    AC0              ; RESTORE AC0
106D          ;
106D B8B2 A          ST      AC2,@AISV2       ; SAVE ACCUMULATORS
106E 88B2 A          LD      AC2,AISV2
106F A2FE A          ST      AC0,DM2(AC2)
1070 A6FF A          ST      AC1,DM1(AC2)
1071 AF01 A          ST      AC3,D1(AC2)
1072          ;
1072 8CE7 A          LD      AC3,ARTC
1073 4C00 A          LI      AC0,C0
1074 0643 A          ROUT    FC43             ; STOP RTC
1075 0980 A          PFLG    INEN             ; DISABLE INTERRUPTS
1076          ;
1076 B8B6 A          LD      AC2,AISTK4       ; SAVE STACK 5 - 16
1077 4D03 A          LI      AC1,C3
1078 4700 A ESTK1:   PULL    AC3
1079 AE01 A          ST      AC3,D1(AC2)
107A 4700 A          PULL    AC3
107B AE02 A          ST      AC3,D2(AC2)
107C 4700 A          PULL    AC3
107D AE03 A          ST      AC3,D3(AC2)
107E 4700 A          PULL    AC3
107F AE04 A          ST      AC3,D4(AC2)
1080 4A04 A          AISZ    AC2,C4
1081 49FF A          AISZ    AC1,CM1
1082 21F5 A          JMP     ESTK1
1083          ;
1083 88B2 A          LD      AC2,AISV2        ; RESTORE ACCUMULATORS
1084 82FE A          LD      AC0,DM2(AC2)
1085 86FF A          LD      AC1,DM1(AC2)
1086 8E01 A          LD      AC3,D1(AC2)
1087 98B2 A          LD      AC2,@AISV2
1088          ;
1088 2C2F A          JSR     @AEADC           ; AUTODISCONNECT ROUTINE, NO RETURN
1089          ;
1089 2100 A          JMP     .+1              ; THIS INST SHOULD NEVER BE REACHED
108A 21FE A          JMP     .-1
108B          ;
108B          ;                  ... DETERMINE SOURCE OF INTERRUPT ...
108B          EGIS3:
108B          ;
108B          ;                  ... RTC IS THE ONLY INTERRUPT PRESENTLY ASSIGNED ...
108B          ;
108B          ;                  ... REAL TIME CLOCK INTERRUPT SERVICING ROUTINE ...
108B 889A A ERTC:    LD      AC2,AEBUF        ; CHECK FOR COMPLETION OF PREVIOUS
108C 8200 A          LD      AC0,D0(AC2)      ;    RTC COMPUTE SEQUENCE
108D 110A A          BOC     CCEZ,ERTC1       ; BRANCH IF NO ERROR
```

```
108E          ;
108E 889F A          LD      AC2,AEERR
108F A202 A          ST      AC0,D2(AC2)      ; STORE RTC COMPUTE FLAG, EERRO
1090 8201 A          LD      AC0,D1(AC2)
1091 4801 A          AISZ    AC0,C1           ; INCREMENT ERROR COUNTER
1092 2100 A          JMP     .+1
1093 A201 A          ST      AC0,D1(AC2)
1094 8200 A          LD      AC0,D0(AC2)
1095 0720 A          SETBIT  D0               ; SET ERROR FLAG BIT 0
1096 A200 A          ST      AC0,D0(AC2)
1097 2425 A          JMP     @AECER           ; RTC ERROR RESPONSE ROUTINE
1098          ;
1098 4C01 A ERTC1:   LI      AC0,C1           ; SET RTC COMPUTE FLAG
1099 A200 A          ST      AC0,D0(AC2)
109A          ;                                ... INITIALIZE STACK FOR RTC SEQUENCE ...
109A          ;
109A 4FFF A          LI      AC3,CM1
109B 4300 A          PUSH    AC3
109C 4300 A          PUSH    AC3              ; PUSH TWO PROTECTION WORDS OF ALL ONES
109D          ;
109D 2C26 A          JSR     @AESCH           ; CALL SCHEDULING ROUTINE
109E          ;
109E 4700 A          PULL    AC3              ; PULL TWO PROTECTION WORDS FROM STACK
109F 4700 A          PULL    AC3
10A0          ;
10A0          ;                                ... ENABLE OTHER INTERRUPTS ...
10A0          ;
10A0          ;                                ... NO OTHER INTERRUPTS PRESENTLY ASSIGNED ...
10A0 889A A          LD      AC2,AEBUF        ; RESET RTC COMPUTE FLAG
10A1 4C00 A          LI      AC0,C0
10A2 A200 A          ST      AC0,D0(AC2)
10A3          ;
10A3          ;                                ... RESTORE COMPUTATIONAL ENVIRONMENT ...
10A3 90AD A          LD      AC0,@AESVS       ; RESTORE SELECT FLAG
10A4 1102 A          BOC     CCEZ,EGIS4
10A5 0A00 A          SFLG    SELF
10A6 2101 A          JMP     .+2
10A7 0A80 A EGIS4:   PFLG    SELF
10A8          ;
10A8          ;
10A8 4D04 A          LI      AC1,C4           ; RESTORE CPU STACK
10A9 89AD A          LD      AC2,AESVS        ; STACK PRECEDES SELECT FLAG WORD
10AA 4AFC A EGIS5:   AISZ    AC2,DM4          ; DECREMENT SAVE AREA ADDRESS BY 4
10AB 8E03 A          LD      AC3,D3(AC2)      ; PUSH STACK FROM BOTTOM TO TOP
10AC 4300 A          PUSH    AC3
10AD 8E02 A          LD      AC3,D2(AC2)
10AE 4300 A          PUSH    AC3
10AF 8E01 A          LD      AC3,D1(AC2)
10B0 4300 A          PUSH    AC3
10B1 8E00 A          LD      AC3,D0(AC2)
10B2 4300 A          PUSH    AC3
10B3 49FF A          AISZ    AC1,CM1          ; DECREMENT PASS COUNTER, SKIP IF DONE
```

```
709 10B4 21F5 A          JMP     EGIS5
710 10B5          ;
711 10B5 90AC A          LD      AC0,@AESVF      ; RESTORE RALU FLAGS
712 10B6 5400 A          XCHRS   AC0
713 10B7 0280 A          PULLF
714 10B8 4000 A          PUSH    AC0
715 10B9          ;
716 10B9          ;                                RESTORE ACCUMULATORS
717 10B9 88AB A          LD      AC2,AESV2
718 10BA 82FE A          LD      AC0,DM2(AC2)
719 10BB 86FF A          LD      AC1,DM1(AC2)
720 10BC 8E01 A          LD      AC3,D1(AC2)
721 10BD 98AB A          LD      AC2,@AESV2
722 10BE          ;
723 10BE 0100 A          RTI                     ; RETURN AND ENABLE INTERRUPTS
724 10BF          ;
725 10BF          ;                  ... RTC ERROR, PREVIOUS SEQUENCE INCOMPLETE ...
726 10CF          ECER:
727 10BF 2C2F A          JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
728 10C0          ;
729 10C0 2100 A          JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
730 10C1 21FE A          JMP     .-1
```

```
10C2                .LIST   LER
10C2        ;
10C2        ;                   ... COMPUTER CONTROL UNIT INITIALIZATION ...
10C2        ECCU:
10C2        ;
10C2        ;                   ... TEST FOR CCU IN SYSTEM ...
10C2 1C01 A         BOC     CCCU,.+2        ; BRANCH IF CCU IS IN SYSTEM
10C3 2115 A         JMP     ECCUR           ; JUMP IF CCU NOT IN SYSTEM
10C4        ;
10C4 9115 A         LD      AC0,@ACCUI      ; CHECK FIRST INST IN CCU ROUTINE
10C5 F115 A         SKNE    AC0,ECCUI       ; SKIP IF CCU NOT IN SYSTEM
10C6 2101 A         JMP     .+2
10C7 2111 A         JMP     ECCUR           ; JUMP TO RETURN
10C8        ;
10C8        ;                   ... INITIALIZE CCU RAM ...
10C8 4C00 A         LI      AC0,C0
10C9 4D30 A         LI      AC1,C48         ; LENGTH OF CCU RAM = 48
10CA 8913 A         LD      AC2,ACWAP       ; INITIALIZE ADDRESS POINTER
10CB        ;
10CB A2F4 A ECCU1:  ST      AC0,DM12(AC2)   ; CLEAR RAM
10CC 4A01 A         AISZ    AC2,C1          ; INCREMENT ADDRESS POINTER
10CD 49FF A         AISZ    AC1,CM1         ; DECREMENT COUNTER, SKIP IF DONE
10CE 21FC A         JMP     ECCU1           ; CONTINUE LOOP
10CF        ;
10CF 890E A         LD      AC2,ACWAP       ; INITIALIZE ADDRESS POINTER
10D0 7AF9 A         ISZ     DM7(AC2)        ; SET STATE WORD = +1
10D1 7E0C A         DSZ     D12(AC2)        ; SET BREAKPOINT ADDRESS 0 = -1
10D2 7E0E A         DSZ     D14(AC2)        ; SET BREAKPOINT ADDRESS 1 = -1
10D3        ;
10D3 810B A         LD      AC0,AECCUR      ; RETURN ADDRESS FROM CCU ROUTINE
10D4 A203 A         ST      AC0,D3(AC2)     ; STORE IN TEMPORARY PC LOC IN CCU RAM
10D5        ;
10D5 8107 A         LD      AC0,ACRAM       ; BASE ADDRESS OF CCU RAM WORK AREA
10D6 A2FA A         ST      AC0,CM6(AC2)    ; INITIALIZE PUSH/PULL POINTER
10D7        ;
10D7 4C00 A         LI      AC0,C0          ; AC0 = 0, INITIAL DISPLAY
10D8 2503 A         JMP     @ACDISP         ; TO CCU ROUTINE, RETURN IS TO ECCUR
10D9        ;
10D9 0200 A ECCUR:  RTS     DO              ; RETURN FROM ECCU
10DA        ;
10DA        ;                   ... LOCAL DATA ...
10DA EF00 A ACCUI:  .WORD   X'EF00          ; LOC OF FIRST INST IN CCU ROUTINE
10DB 86F7 A ECCUI:  .WORD   X'86F7          ; FIRST INST IN CCU ROUTINE
10DC EF84 A ACDISP: .WORD   X'EF84          ; CCU ROM + X'84
10DD ECD0 A ACRAM:  .WORD   X'ECD0          ; FIRST LOC IN CCU RAM WORK AREA
10DE ECDC A ACWAP:  .WORD   X'ECDC          ; CCU RAM WORK AREA + X'C
10DF 10D9 A AECCUR: .WORD   ECCUR           ; RETURN ADDRESS FROM CCU INITIALIZATION
```

```
10E0                .LIST   LER
10E0        ;
10E0        ;                   ... INITIALIZATION RAM ADDRESS TEST ...
10E0        EMAD:
10E0        ;
10E0        ;                   ... TEST 4K STATIC RAM ...
10E0        ;                       (CAN ONLY TEST ADDRESS -NOT- USED BY DATA BASE)
10E0 80D6 A         LD      AC0,ADEND       ; START ADDRESS (PAST DATA BASE)
10E1 B0D4 A         ST      AC0,@ASSAD
10E2 80F2 A         LD      AC0,H3FFF       ; END ADDRESS
10E3 B0D5 A         ST      AC0,@ASEAD
10E4        ;
10E4 2C5A A         JSR     @ASMAD
10E5 2109 A         JMP     EMADE           ; ERROR RETURN, (AC2) = CONTENTS
10E6                                        ;               (AC3) = ADDRESS
10E6 2100 A         JMP     .+1             ; NO ERROR RETURN
10E7        ;
10E7        ;                   ... TEST CCU RAM ON IMP/16C CARD ...
10E7        ;                       ( CAN ONLY TEST ADDRESSES -NOT- USED BY CCU)
10E7 80F3 A         LD      AC0,HEC00       ; START ADDRESS
10E8 B0D4 A         ST      AC0,@ASSAD
10E9 80F4 A         LD      AC0,HECCF       ; END ADDRESS
10EA B0D5 A         ST      AC0,@ASEAD
10EB        ;
10EB 2C5A A         JSR     @ASMAD
10EC 2102 A         JMP     EMADE           ; ERROR RETURN, (AC2) = CONTENTS
10ED                                        ;               (AC3) = ADDRESS
10ED 2100 A         JMP     .+1             ; NO ERROR RETURN
10EE        ;
10EE 2103 A         JMP     EMADR
10EF        ;
10EF 2C2F A EMADE:  JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
10F0        ;
10F0 2100 A         JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
10F1 21FE A         JMP     .-1
10F2        ;
10F2 0200 A EMADR:  RTS     D0              ; RETURN FROM EMAD
10F3        ;
```

```
10F3                .LIST   LER
10F3          ;
10F3          ;                 ... RTC SCHEDULING ROUTINE ...
10F3          ESCH:
10F3 8CE7 A         LD      AC3,ARTC
10F4 4C00 A         LI      AC0,C0          ; CLEAR AC0
10F5 0641 A         ROUT    FC41            ; RESET RTC INTERRUPT PULSE
10F6          ;
10F6 0900 A         SFLG    INEN              ; ENABLE INTERRUPTS
10F7          ;
10F7 1E01 A         BOC     CCCF,.+2        ; BRANCH IF COTF JUMP CONDITION SET
10F8 2101 A         JMP     .+2
10F9 2C3D A         JSR     @ACOTF          ; CALL CATCH ON THE FLY ROUTINE
10FA          ;
10FA 2C39 A         JSR     @AXRTC
10FB 2C3C A         JSR     @ABRTC
10FC 2C27 A         JSR     @AEDIS
10FD 2C28 A         JSR     @AECOM
10FE 2C29 A         JSR     @AESYN
10FF 2C2A A         JSR     @AEMIC
1100 2C2B A         JSR     @AETIM
1101 2C2E A         JSR     @AEDET
1102 2C30 A         JSR     @APMON
1103 2C35 A         JSR     @ATPEX
1104 2C3B A         JSR     @AXSDC
1105 2C2D A         JSR     @AESDW
1106 2C40 A         JSR     @AEPPM
1107          ;
1107 2105 A         JMP     ERTS
1108          ;
1108 8905 A         LD      AC2,EDLAY
1109 5C76 A         SHL     AC0,D118
110A 5C74 A         SHL     AC0,D116
110B 4A01 A         AISZ    AC2,C1
110C 21FC A         JMP     .-3
110D          ;
110D 0200 A ERTS:   RTS     D0
110E          ;                                VARIABLE DELAY
110E FFCE A EDLAY:  .WORD   -50
110F          ;
110F          ;
```

```
865 110F                .LIST   LER
866 110F        ;
867 110F        ;                   ... AUTODISCONNECT ROUTINE ...
868 110F        EADC:
869 110F B8B7 A         ST      AC2,@AEDSV2     ; SAVE ACCUMULATORS
870 1110 88B7 A         LD      AC2,AEDSV2
871 1111 A2FE A         ST      AC0,DM2(AC2)
872 1112 A6FF A         ST      AC1,DM1(AC2)
873 1113 AE01 A         ST      AC3,D1(AC2)
874 1114        ;
875 1114 8CE7 A         LD      AC3,ARTC
876 1115 4C00 A         LI      AC0,C0
877 1116 0643 A         ROUT    FC43            ; STOP RTC
878 1117 0641 A         ROUT    FC41            ; ASSURE RTC INTERRUPT PULSE RESET
879 1118 0980 A         PFLG    INEN            ; DISABLE INTERRUPTS
880 1119        ;
881 1119 4400 A         PULL    AC0             ; SAVE RALU FLAGS
882 111A 0080 A         PUSHF
883 111B 5400 A         XCHRS   AC0
884 111C B0B8 A         ST      AC0,@AEDSVF
885 111D        ;
886 111D 4D04 A         LI      AC1,C4          ; SAVE CPU STACK
887 111E 88D8 A         LD      AC2,AEDSVF      ; STACK FOLLOWS RALU FLAG WORD
888 111F 4700 A EADC1:  PULL    AC3             ; PULL STACK FROM TOP TO BOTTOM
889 1120 AF01 A         ST      AC3,D1(AC2)
890 1121 4700 A         PULL    AC3
891 1122 AF02 A         ST      AC3,D2(AC2)
892 1123 4700 A         PULL    AC3
893 1124 AE03 A         ST      AC3,D3(AC2)
894 1125 4700 A         PULL    AC3
895 1126 AE04 A         ST      AC3,D4(AC2)
896 1127 4A04 A         AISZ    AC2,D4          ; INCREMENT SAVE AREA ADDRESS BY 4
897 1128 49FF A         AISZ    AC1,CM1         ; DECREMENT PASS COUNTER, SKIP IF DONE
898 1129 21F5 A         JMP     EADC1
899 112A        ;
900 112A        ;
901 112A 4C01 A         LI      AC0,C1          ; SAVE SELECT FLAG STATE
902 112B 58EF A         ROR     AC0,D17
903 112C 1303 A         BOC     CCB0,EADC2      ; BRANCH IF SELECT FLAG SET
904 112D 4C00 A         LI      AC0,C0
905 112E B0B9 A         ST      AC0,@AEDSVS     ; SELECT FLAG = 0
906 112F 2102 A         JMP     EADC3
907 1130 4C01 A EADC2:  LI      AC0,C1
908 1131 B0B9 A         ST      AC0,@AEDSVS     ; SELECT FLAG = 1
909 1132        ;
910 1132        EADC3:
911 1132        ;
912 1132 2C2D A         JSR     @AESDW          ; CONSTRUCT STATUS DISPLAY WORD
913 1133        ;
914 1133 906B A         LD      AC0,@AXBUF      ; FETCH SDW
915 1134 0751 A         SKBIT   D1              ; CHECK EXTERNAL DISCONNECT BIT
916 1135 2101 A         JMP     .+2             ; JUMP IF NOT EXTERNAL
```

```
1136 2104 A          JMP     EADC4          ; JUMP IF EXTERNAL
1137 0720 A          SETBIT  DO             ; SET DISCONNECT BIT
1138 B06B A          ST      AC0,@AXBUF
1139 8CE3 A          LD      AC3,ATCP       ; OUTPUT CONTROL PULSE FOR
113A 0643 A          ROUT    FC43           ;    INTERNAL DISCONNECT TO TCP
113B        ;
113B 8CE4 A EADC4:   LD      AC3,ASDU       ; OUTPUT SDW TO SDU
113C 906B A          LD      AC0,@AXBUF     ; FETCH SDW
113D 0601 A          ROUT    FC1
113E        ;
113E        ;
113E 3081 A          NOP                    ; TERMINAL WAIT LOOP, NO RETURN
113F 3081 A          NOP
1140 21FD A          JMP     .-2
1141 0000 A          HALT                   ; THIS HALT SHOULD NEVER BE REACHED
1142 0000 A          HALT
```

```
1143                .LIST   LER
1143        ;
1143        ;                   ... TIME OUT FOR DEBUG ROUTINE ...
1143        ;*****************************************************************************
1143        ;
1143        COTF:
1143 B8AF A         ST      AC2,@ACSV2              ; SAVE ACCUMULATORS
1144 88AF A         LD      AC2,ACSV2
1145 A2FE A         ST      AC0,DM2(AC2)
1146 A6FF A         ST      AC1,DM1(AC2)
1147 AE01 A         ST      AC3,D1(AC2)
1148        ;
1148 4400 A         PULL    AC0                     ; SAVE RALU FLAGS
1149 0080 A         PUSHF
114A 5400 A         XCHRS   AC0
114B B0B0 A         ST      AC0,@ACSVF
114C        ;
114C 4D04 A         LI      AC1,C4                  ; SAVE CPU STACK
114D 88B0 A         LD      AC2,ACSVF               ; STACK FOLLOWS RALU FLAG WORD
114E 4700 A COTF1:  PULL    AC3                     ; PULL STACK FROM TOP TO BOTTOM
114F AE01 A         ST      AC3,D1(AC2)
1150 4700 A         PULL    AC3
1151 AE02 A         ST      AC3,D2(AC2)
1152 4700 A         PULL    AC3
1153 AF03 A         ST      AC3,D3(AC2)
1154 4700 A         PULL    AC3
1155 AF04 A         ST      AC3,D4(AC2)
1156 4A04 A         AISZ    AC2,D4                  ; INCREMENT SAVE AREA ADDRESS BY 4
1157 49FF A         AISZ    AC1,CM1                 ; DECREMENT PASS COUNTER, SKIP IF DONE
1158 21F5 A         JMP     COTF1
1159        ;
1159        ;
1159 4C01 A         LI      AC0,C1                  ; SAVE SELECT FLAG STATE
115A 58EF A         ROR     AC0,D17
115B 1303 A         BOC     CC30,COTF2              ; BRANCH IF SELECT FLAG SET
115C 4C00 A         LI      AC0,C0
115D B0B1 A         ST      AC0,@ACSVS              ; SELECT FLAG = 0
115E 2102 A         JMP     COTF3
115F 4C01 A COTF2:  LI      AC0,C1
1160 B0B1 A         ST      AC0,@ACSVS              ; SELECT FLAG = 1
1161        ;
1161        ;                                       STOP THE CLOCK
1161        COTF3:
1161 8CE7 A         LD      AC3,ARTC
1162 4C00 A         LI      AC0,C0
1163 0643 A         ROUT    FC43                    ; STOP RTC
1164 0641 A         ROUT    FC41                    ; RESET RTC INTERRUPT PULSE
1165        ;
1165        ;-----------------------------------------------------------------------------
1165 3081 A         NOP                             ; INSERT DEBUG HALT OR SNAP HERE
1166 3081 A         NOP
1167 2CA9 A         JSR     @AERAM                  ; FIRST LOCATION IN FREE RAM
```

```
 987 1168 3081 A          NOP
 988 1169 3081 A          NOP
 989 116A 3081 A          NOP
 990 116B          ;-----------------------------------------------------------------
 991 116B          ;
 992 116B          ;                                    START THE CLOCK
 993 116B 8CE7 A          LD       AC3,ARTC
 994 116C 4C00 A          LI       AC0,C0
 995 116D 0642 A          ROUT     FC42                    ; START RTC
 996 116E          ;
 997 116E          ;                  ... RESTORE COMPUTATIONAL ENVIRONMENT ...
 998 116E          ;
 999 116E 90B1 A          LD       AC0,@ACSVS        ; RESTORE SELECT FLAG
1000 116F 1102 A          BOC      CCEZ,COTF4
1001 1170 0A00 A          SFLG     SELF
1002 1171 2101 A          JMP      .+2
1003 1172 0A80 A COTF4:   PFLG     SELF
1004 1173         ;
1005 1173         ;
1006 1173 4004 A          LI       AC1,C4            ; RESTORE CPU STACK
1007 1174 88B1 A          LD       AC2,ACSVS         ; STACK PRECEDES SELECT FLAG WORD
1008 1175 4AFC A COTF5:   AISZ     AC2,DM4           ; DECREMENT SAVE AREA ADDRESS BY 4
1009 1176 8E03 A          LD       AC3,D3(AC2)       ; PUSH STACK FROM BOTTOM TO TOP
1010 1177 4300 A          PUSH     AC3
1011 1178 8E02 A          LD       AC3,D2(AC2)
1012 1179 4300 A          PUSH     AC3
1013 117A 8E01 A          LD       AC3,D1(AC2)
1014 117B 4300 A          PUSH     AC3
1015 117C 8E00 A          LD       AC3,D0(AC2)
1016 117D 4300 A          PUSH     AC3
1017 117E 49FF A          AISZ     AC1,CM1           ; DECREMENT PASS COUNTER, SKIP IF DONE
1018 117F 21F5 A          JMP      COTF5
1019 1180         ;
1020 1180 90D0 A          LD       AC0,@ACSVF        ; RESTORE RALU FLAGS
1021 1181 5400 A          XCHRS    AC0
1022 1182 0280 A          PULLF
1023 1183 4000 A          PUSH     AC0
1024 1184         ;
1025 1184         ;                                    RESTORE ACCUMULATORS
1026 1184 88AF A          LD       AC2,ACSV2
1027 1185 82FE A          LD       AC0,DM2(AC2)
1028 1186 86FF A          LD       AC1,DM1(AC2)
1029 1187 8E01 A          LD       AC3,D1(AC2)
1030 1188 98AF A          LD       AC2,@ACSV2
1031 1189         ;
1032 1189 0200 A          RTS      D0                ; RETURN FROM COTF
```

```
 1033 118A                .PAGE
 1034 118A                .ASECT
 1035 118A 1200 A         .=X'1200
 1036 1200        ;
```

```
1200                .LIST   LES
1200        ;
1200        ;                   ... MONITOR FOR DISCONNECT ...
1200        EDIS:
1200        ;
1200        ;                   ... TEST PROCESSOR STATUS FLAGS ...
1200 0744 A         SKSTF   D4              ; SKIP IF RAM ADDRESS TEST ERROR
1201 2101 A         JMP     .+2
1202 210C A         JMP     EDISH
1203        ;
1203        ;                   ... RESERVE TESTS FOR ADDITIONAL STATUS FLAGS ...
1203 0745 A         SKSTF   D5
1204 2101 A         JMP     .+2
1205 2109 A         JMP     EDISH
1206        ;
1206 0746 A         SKSTF   D6
1207 2101 A         JMP     .+2
1208 2106 A         JMP     EDISH
1209        ;
1209 0747 A         SKSTF   D7
120A 2101 A         JMP     .+2
120B 2103 A         JMP     EDISH
120C        ;
120C 9069 A         LD      AC0,@ABDIN      ; DISCRETE INPUT WORD
120D 075F A         SKBIT   D15             ; SKIP IF DISCONNECT BIT IS SET
120E 0200 A         RTS     D0
120F        ;
120F 2C2F A EDISH:  JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
1210        ;
1210 2100 A         JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
1211 21FE A         JMP     .-1
1212        ;
```

```
1212                    .LIST   LES
1212            ;
1212            ;                   ... MONITOR COMMAND FEEDBACK DATA FROM LDU ...
1212            ECOM:
1212 4C00 A             LI      AC0,C0
1213 B0A0 A             ST      AC0,@AECMF      ; CLEAR CURRENT RTC INTERVAL ERROR FLAG
1214            ;
1214            ;                   ... COMMAND FEEDBACK TEST FOR MAIN STREET ...
1214 886B A             LD      AC2,AXBUF
1215 8201 A             LD      AC0,D1(AC2)     ; XOCW1 OUTPUT COMMAND
1216 60F2 A             AND     AC0,H3FFF       ; MASK OFF DON'T CARE BITS 15,14
1217 8868 A             LD      AC2,ABBUF
1218 8601 A             LD      AC1,D1(AC2)     ; BCR1 FEEDBACK COMMAND
1219 60F2 A             AND     AC0,H3FFF       ; MASK OFF DON'T CARE BITS 15,14
121A 3482 A             RXOR    AC1,AC0         ; COMPARE, AC1 UNALTERED
121B 110B A             BOC     CCEZ,ECOM1      ; BRANCH IF IDENTICAL (AC0=0)
121C            ;
121C            ;                   ... MAIN STREET COMMAND FEEDBACK ERROR ...
121C 889F A             LD      AC2,AEERR
121D A604 A             ST      AC1,D4(AC2)     ; STORE BCR1 IN ERROR BUFFER, EERR1
121E 8203 A             LD      AC0,D3(AC2)     ; FETCH ERROR COUNTER AND INCREMENT
121F 4801 A             AISZ    AC0,C1
1220 2100 A             JMP     .+1
1221 A203 A             ST      AC0,D3(AC2)
1222 8200 A             LD      AC0,D0(AC2)
1223 0721 A             SETBIT  D1              ; SET ERROR FLAG BIT 1
1224 A200 A             ST      AC0,D0(AC2)
1225 4C01 A             LI      AC0,C1
1226 B0A0 A             ST      AC0,@AECMF      ; SET CURRENT RTC INTERVAL ERROR FLAG
1227            ;
1227            ;                   ... COMMAND FEEDBACK TEST FOR CROSS STREET ...
1227            ECOM1:
1227 886B A             LD      AC2,AXBUF
1228 8202 A             LD      AC0,D2(AC2)     ; XOCW2 OUTPUT COMMAND
1229 60F2 A             AND     AC0,H3FFF       ; MASK OFF DON'T CARE BITS 15,14
122A 8868 A             LD      AC2,ABBUF
122B 8602 A             LD      AC1,D2(AC2)     ; BCR2 FEEDBACK COMMAND
122C 60F2 A             AND     AC0,H3FFF       ; MASK OFF DON'T CARE BITS 15,14
122D 3482 A             RXOR    AC1,AC0         ; COMPARE, AC1 UNALTERED
122E 110C A             BOC     CCEZ,ECOM2      ; BRANCH IF IDENTICAL (AC0=0)
122F            ;
122F            ;                   ... CROSS STREET COMMAND FEEDBACK ERROR ...
122F 889F A             LD      AC2,AEERR
1230 A606 A             ST      AC1,D6(AC2)     ; STORE BCR2 IN ERROR BUFFER, EERR2
1231 8205 A             LD      AC0,D5(AC2)     ; FETCH ERROR COUNTER AND INCREMENT
1232 4801 A             AISZ    AC0,C1
1233 2100 A             JMP     .+1
1234 A205 A             ST      AC0,D5(AC2)
1235 8200 A             LD      AC0,D0(AC2)
1236 0722 A             SETBIT  D2              ; SET ERROR FLAG BIT 2
1237 A200 A             ST      AC0,D0(AC2)
1238 4C02 A             LI      AC0,C2
```

```
1125 1239 B0A0 A        ST      AC0,@AECMF      ; SET CURRENT RTC INTERVAL ERROR FLAG
1126 123A 2103 A        JMP     ECOMR
1127 123B         ;
1128 123B 90A0 A ECOM2: LD      AC0,@AECMF      ; TEST CURRENT RTC INTERVAL ERROR FLAG
1129 123C 1501 A        BOC     CCNZ,ECOMR      ; BRANCH IF ERROR
1130 123D         ;
1131 123D 2C3A A        JSR     @AXWDT          ; RESET WATCHDOG TIMER IF NO ERROR
1132 123E         ;
1133 123E 0200 A ECOMR: RTS     D0              ; RETURN FROM ECOM
1134 123F         ;
```

```
1137 123F                .LIST   LES
1138 123F        ;
1139 123F        ;                  ... LOOP MASTER SYNCHRONIZATION ROUTINE ...
1140 123F        ESYN:
1141 123F 90A2 A         LD      AC0,@AER123      ; FETCH RESET SELECT DATUM
1142 1240 1125 A         BOC     CCEZ,ESYN4A      ; BRANCH IF NO RESET IS HIGH (PULSE)
1143 1241 4D02 A         LI      AC1,C2
1144 1242 3482 A         RXOR    AC1,AC0
1145 1243 1113 A         BOC     CCEZ,ESYN1       ; BRANCH IF ONLY RESET 1 IS HIGH
1146 1244 90A2 A         LD      AC0,@AER123      ; RESTORE RESET SELECT DATUM
1147 1245 4D04 A         LI      AC1,C4
1148 1246 3482 A         RXOR    AC1,AC0
1149 1247 1114 A         BOC     CCEZ,ESYN2       ; BRANCH IF ONLY RESET 2 IS HIGH
1150 1248 90A2 A         LD      AC0,@AER123      ; RESTORE RESET SELECT DATUM
1151 1249 4D08 A         LI      AC1,C8
1152 124A 3482 A         RXOR    AC1,AC0
1153 124B 1115 A         BOC     CCEZ,ESYN3       ; BRANCH IF ONLY RESET 3 IS HIGH
1154 124C 889F A         LD      AC2,AEERR        ; ERROR, MORE THAN ONE RESET IS HIGH
1155 124D 90A2 A         LD      AC0,@AER123      ; RESTORE RESET SELECT DATUM
1156 124E A208 A         ST      AC0,D8(AC2)      ; STORE IN ERROR BUFFER, EERR3
1157 124F 8207 A         LD      AC0,D7(AC2)      ; FETCH ERROR COUNTER AND INCREMENT
1158 1250 4801 A         AISZ    AC0,C1
1159 1251 2100 A         JMP     .+1
1160 1252 A207 A         ST      AC0,D7(AC2)
1161 1253 8200 A         LD      AC0,D0(AC2)
1162 1254 0723 A         SETBIT  D3               ; SET ERROR FLAG BIT 3
1163 1255 A200 A         ST      AC0,D0(AC2)
1164 1256 0200 A         RTS     D0
1165 1257        ;
1166 1257        ESYN1:                            ; RESET 1
1167 1257 2921 A         JSR     ESYA             ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1168 1258 8201 A         LD      AC0,D1(AC2)      ; RESET 1 POSITION (RTC UNITS)
1169 1259 D205 A         SUB     AC0,D5(AC2)      ; COMPARE WITH SYNCH DIAL TIMER
1170 125A 111D A         BOC     CCEZ,ESYN6       ; BRANCH IF DIAL IS AT RESET POINT
1171 125B 210B A         JMP     ESYN4
1172 125C        ;
1173 125C        ESYN2:                            ; RESET 2
1174 125C 291C A         JSR     ESYA             ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1175 125D 8202 A         LD      AC0,D2(AC2)      ; RESET 2 POSITION (RTC UNITS)
1176 125E D205 A         SUB     AC0,D5(AC2)      ; COMPARE WITH SYNCH DIAL TIMER
1177 125F 1118 A         BOC     CCEZ,ESYN6       ; BRANCH IF DIAL IS AT RESET POINT
1178 1260 2106 A         JMP     ESYN4
1179 1261        ;
1180 1261        ESYN3:                            ; RESET 3
1181 1261 2917 A         JSR     ESYA             ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1182 1262 8203 A         LD      AC0,D3(AC2)      ; RESET 3 POSITION (RTC UNITS)
1183 1263 D205 A         SUB     AC0,D5(AC2)      ; COMPARE WITH SYNCH DIAL TIMER
1184 1264 1113 A         BOC     CCEZ,ESYN6       ; BRANCH IF DIAL IS AT RESET POINT
1185 1265 2101 A         JMP     ESYN4
1186 1266        ;
1187 1266 2912 A ESYN4A: JSR     ESYA             ; COMPUTE ADDRESS OF DATA BUFFER (AC2)
1188 1267        ESYN4:                            ; INCREMENT SYNCH DIAL TIMER
```

```
1267 7A05 A          ISZ     D5(AC2)
1268 2100 A          JMP     .+1
1269 8200 A          LD      AC0,D0(AC2)      ; FETCH CYCLE LENGTH
126A E205 A          SKG     AC0,D5(AC2)      ; SKIP IF BEFORE END OF CYCLE
126B 2101 A          JMP     .+2
126C 2105 A          JMP     ESYN5
126D 4C00 A          LI      AC0,C0
126E A205 A          ST      AC0,D5(AC2)      ; ZERO SYNCH DIAL TIMER
126F 4C01 A          LI      AC0,C1
1270 B09D A          ST      AC0,@AEPST       ; SET START PULSE
1271 2106 A          JMP     ESYN6
1272          ;
1272          ESYN5:
1272 8204 A          LD      AC0,D4(AC2)      ; FETCH END OF START PULSE
1273 E205 A          SKG     AC0,D5(AC2)      ; SKIP IF BEFORE END OF START PULSE
1274 2101 A          JMP     .+2
1275 2102 A          JMP     ESYN6
1276 4C00 A          LI      AC0,C0
1277 B09D A          ST      AC0,@AEPST       ; RESET START PULSE
1278          ;
1278          ESYN6:
1278 0200 A          RTS     D0               ; RETURN FROM ESYN
1279          ;
1279          ESYA:                            ; COMPUTE BASE ADDRESS OF SYNCH DATA
1279                                           ;    FOR OPERATING DIAL, (AC2)
1279 4C06 A          LI      AC0,C6           ; BUFFER LENGTH = 6
127A 88A1 A          LD      AC2,AESYD        ; BASE ADDRESS FOR DIAL 1
127B 9C83 A          LD      AC3,@ATCDS       ; CURRENT DIAL SELECTION
127C 4BFF A ESYA1:   ISZ     AC3,CM1          ; SKIP IF BASE ADDRESS COMPUTED
127D 2101 A          JMP     .+2
127E 2102 A          JMP     ESYA2
127F 3200 A          RADD    AC0,AC2          ; INCREMENT BY BUFFER LENGTH
1280 21FB A          JMP     ESYA1
1281          ESYA2:
1281 0200 A          RTS     D0               ; RETURN FROM ESYA
1282          ;
```

```
1282                    .LIST   LES
1282            ;
1282            ;               ... MONITOR INTERCONNECT DATA (R1,R2,R3,D2,D3) ...
1282            EMIC:
1282            ;               ... NOISE OR TRANSITION FILTER ON D2,D3 ...
1282            ;
1282 9069 A             LD      AC0,@ABDIN      ; ACCESS BDIN
1283 611E A             AND     AC0,H0030       ; EXTRACT D2,D3 TO GET NEW EDN0
1284 8CA5 A             LD      AC3,AEDN0       ; PAST VALUE BUFFER BASE ADDRESS
1285 8700 A             LD      AC1,D0(AC3)     ; UPDATE PAST VALUES
1286 8801 A             LD      AC2,D1(AC3)
1287 A300 A             ST      AC0,D0(AC3)     ; STORE IN BUFFER, EDN0, 0TH PAST VALUE
1288 A701 A             ST      AC1,D1(AC3)     ;                  EDN1, 1ST PAST VALUE
1289 A802 A             ST      AC2,D2(AC3)     ;                  EDN2, 2ND PAST VALUE
128A 3482 A             RXOR    AC1,AC0
128B 1505 A             BOC     CCNZ,EMIC1      ; BRANCH IF EDN0.NE.EDN1
128C 8300 A             LD      AC0,D0(AC3)     ; RESTORE EDN0
128D 3882 A             RXOR    AC2,AC0
128E 1502 A             BOC     CCNZ,EMIC1      ; BRANCH IF EDN0.NE.EDN2
128F 8300 A             LD      AC0,D0(AC3)     ; RESTORE EDN0
1290 B0A4 A             ST      AC0,@AED23      ; THREE PAST VALUES ARE IDENTICAL
1291                                            ;    UPDATE INTERNAL DIAL SELECT DATUM
1291            EMIC1:
1291            ;               ... NOISE OR TRANSITION FILTER ON R1,R2,R3 ...
1291            ;
1291 9069 A             LD      AC0,@ABDIN      ; ACCESS BDIN
1292 610E A             AND     AC0,H000E       ; EXTRACT R1,R2,R3 TO GET NEW ERN0
1293 8CA3 A             LD      AC3,AERN0       ; PAST VALUE BUFFER BASE ADDRESS
1294 8700 A             LD      AC1,D0(AC3)     ; UPDATE PAST VALUES
1295 8801 A             LD      AC2,D1(AC3)
1296 A300 A             ST      AC0,D0(AC3)     ; STORE IN BUFFER, ERN0, 0TH PAST VALUE
1297 A701 A             ST      AC1,D1(AC3)     ;                  ERN1, 1ST PAST VALUE
1298 A802 A             ST      AC2,D2(AC3)     ;                  ERN2, 2ND PAST VALUE
1299 3482 A             RXOR    AC1,AC0
129A 1505 A             BOC     CCNZ,EMIC2      ; BRANCH IF ERN0.NE.ERN1
129B 8300 A             LD      AC0,D0(AC3)     ; RESTORE ERN0
129C 3882 A             RXOR    AC2,AC0
129D 1502 A             BOC     CCNZ,EMIC2      ; BRANCH IF ERN0.NE.ERN2
129E 8300 A             LD      AC0,D0(AC3)     ; RESTORE ERN0
129F B0A2 A             ST      AC0,@AER123     ; THREE PAST VALUES ARE IDENTICAL
12A0                                            ;    UPDATE INTERNAL RESET SELECT DATUM
12A0            EMIC2:
12A0 0200 A             RTS     D0              ; RETURN FROM EMIC
12A1            ;
12A1            ;               ... LOCAL DATA ...
12A1 000E A     H000E:  .WORD   X'000E          ; BITS 1,2,3 SET
12A2 0030 A     H0030:  .WORD   X'0030          ; BITS 4,5 SET
12A3            ;
```

```
12A3                .LIST   LES
12A3        ;
12A3        ;                   ... EXECUTIVE TIMING ROUTINE ...
12A3        ETIM:
12A3 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
12A4        ;
12A4 9069 A         LD      AC0,@ABDIN          ; FETCH BDIN
12A5 075E A         SKBIT   D14                 ; SKIP IF WAIT CONDITION
12A6 2101 A         JMP     .+2
12A7 2105 A         JMP     ETIM1
12A8                                            ; BYPASS THESE FNS IN THE WAIT STATE
12A8 909B A         LD      AC0,@AETCY          ; TIMING PLAN CYCLE TIMER
12A9 4801 A         AISZ    AC0,C1              ; INCREMENT TIMERS
12AA 2100 A         JMP     .+1
12AB B09B A         ST      AC0,@AETCY          ; STORE CYCLE TIMER
12AC B09C A         ST      AC0,@AETST          ; STORE 'WAIT FOR SYNCH START' TIMER
12AD        ;
12AD 90A6 A ETIM1:  LD      AC0,@AEDMIT         ; INCREMENT DETECTOR MONITORING
12AE 4801 A         AISZ    AC0,C1              ;    INTERVAL TIMER
12AF 2100 A         JMP     .+1
12B0 B0A6 A         ST      AC0,@AEDMIT
12B1        ;
12B1 0200 A ETIMR:  RTS     D0
12B2        ;
```

```
12B2                    .LIST   LES
12B2            ;
12B2            ;                   ... DETECTOR MONITORING ROUTINE ...
12B2            EDET:
12B2 90A6 A             LD      AC0,@AEDMIT     ; FETCH TIMER
12B3 E128 A             SKG     AC0,EDETI       ; SKIP IF DETECTOR MONITORING INTERVAL
12B4                                            ;    HAS BEEN REACHED
12B4 2126 A             JMP     EDETR
12B5            ;
12B5 8C7C A             LD      AC3,AVCVC       ; BASE ADDRESS OF VEHICLE COUNTERS
12B6            ;
12B6 8300 A     EDET1:  LD      AC0,(AC3)       ; FETCH NEXT VEHICLE COUNTER
12B7 E125 A             SKG     AC0,EDETM       ; SKIP IF GT MINIMUM COUNT
12B8 2101 A             JMP     .+2             ; UNDERCOUNTING ERROR
12B9 2119 A             JMP     EDET3           ; NO ERROR
12BA            ;
12BA            ;                   ... UNDERCOUNTING ERROR ...
12BA 3E81 A             RCPY    AC3,AC2         ; COPY CURRENT VEHICLE COUNTER ADDRESS
12BB 847C A             LD      AC1,AVCVC       ; BASE ADDRESS OF VEHICLE COUNTERS
12BC 5101 A             CAI     AC1,C1          ; COMPUTE DETECTOR INDEX
12BD 3600 A             RADD    AC1,AC2
12BE E91F A             SKG     AC2,EDET32      ; SKIP IF INVALID INDEX
12BF 2106 A             JMP     EDET2
12C0            ;
12C0            ;                   ... RECORD DETECTOR INDEX ERROR ...
12C0 3981 A             RCPY    AC2,AC1         ; COPY ERROR DATUM TO AC1
12C1 BCA7 A             ST      AC3,@AEDETT     ; STORE VEHICLE COUNTER ADDRESS IN TEMP
12C2 8C9F A             LD      AC3,AEERR       ; BASE ADDRESS OF ERROR BUFFER
12C3 2C4E A             JSR     @AERE5          ; AC1 HAS ERROR DATUM
12C4 9CA7 A             LD      AC3,@AEDETT     ; RESTORE VEHICLE COUNTER ADDRESS
12C5 210D A             JMP     EDET3
12C6            ;
12C6            ;                   ... RECORD ERROR CURRENT DETECTOR ...
12C6 B8A7 A     EDET2:  ST      AC2,@AEDETT     ; STORE DETECTOR INDEX IN TEMP LOC
12C7 5E01 A             SHL     AC2,D1          ; MPY BY 2 FOR DISP IN ERROR BUFFER
12C8 C8A8 A             ADD     AC2,AEDERR      ; ADD TO BASE ADDRESS
12C9 A201 A             ST      AC0,D1(AC2)     ; STORE UNDERCOUNT VALUE
12CA 8200 A             LD      AC0,D0(AC2)     ; INCREMENT ERROR COUNTER
12CB 4801 A             AISZ    AC0,C1
12CC 2100 A             JMP     .+1
12CD A200 A             ST      AC0,D0(AC2)
12CE            ;
12CE            ;                   ... RECORD GENERAL UNDERCOUNTING ERROR ...
12CE 94A7 A             LD      AC1,@AEDETT     ; TEMP HAS DETECTOR INDEX, ERROR DATUM
12CF BCA7 A             ST      AC3,@AEDETT     ; STORE VEHICLE COUNTER ADDRESS IN TEMP
12D0 8C9F A             LD      AC3,AEERR       ; BASE ADDRESS OF ERROR BUFFER
12D1 2C4D A             JSR     @AERE4          ; AC1 HAS ERROR DATUM
12D2 9CA7 A             LD      AC3,@AEDETT     ; RESTORE VEHICLE COUNTER ADDRESS
12D3            ;
12D3 4C00 A     EDET3:  LI      AC0,C0
12D4 A300 A             ST      AC0,(AC3)       ; ZERO VEHICLE COUNTER
12D5 4B01 A             AISZ    AC3,C1          ; INCREMENT VEHICLE COUNTER ADDRESS
```

```
12D6 2100 A        JMP     .+1
12D7 ED07 A        SKG     AC3,EVCVC        ; SKIP IF ALL VEHICLE COUNTERS PROCESSED
12D8 21DD A        JMP     EDET1
12D9 4C00 A        LI      AC0,C0
12DA B0A6 A        ST      AC0,@AEDMIT      ; ZERO DETECTOR MONITORING INT TIMER
12DB          ;
12DB 0200 A EDETR: RTS     D0               ; RETURN FROM EDET
12DC          ;
12DC          ;                               ... LOCAL DATA ...
12DC 2EE0 A EDETI:  .WORD  12000            ; DET MONITORING INT (0.1 SEC UNITS)
12DD 0001 A EDETM:  .WORD  1                ; MINIMUM VEHICLE COUNT
12DE 001F A EDET32: .WORD  31               ; MAXIMUM DETECTOR INDEX (0 - 31)
12DF 3131 A EVCVC:  .WORD  VCVC+DETS-1      ; ADDRESS OF LAST DETECTOR VEHICLE CTR
12E0                                        ;    (BASE ADDR + NO. DETECTORS - 1)
12E0          ;
```

```
1372 12E0                .LIST   LES
1373 12E0        ;
1374 12E0        ;                   ... CONSTRUCT STATUS DISPLAY WORD ...
1375 12F0        ESDW:
1376 12F0        ;
1377 12F0        ;                   ... BITS 15 TO 11 ...
1378 12F0 90A2 A         LD      AC0,@AER123     ; R1,R2,R3,IN BITS 1 TO 3
1379 12E1 88A4 A         LD      AC2,AED23
1380 12E2 6A00 A         OR      AC0,(AC2)       ; D2,D3 IN BITS 4,5
1381 12E3 0A80 A         PFLG    SELF            ; ASSURE SEL = 0
1382 12E4 58FA A         ROR     AC0,D6          ; R1,R2,R3,D2,D3 IN BITS 11 TO 15
1383 12E5 6132 A         AND     AC0,HF800       ; ASSURE BITS 10 TO 0 RESET
1384 12E6 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
1385 12E7        ;
1386 12E7        ;                   ... BIT 10 ...
1387 12E7 909D A         LD      AC0,@AEPST
1388 12E8 1103 A         BOC     CCEZ,.+4        ; BRANCH IF NO START PULSE
1389 12E9 3481 A         RCPY    AC1,AC0
1390 12EA 072A A         SETBIT  D10             ; START PULSE
1391 12EB 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
1392 12EC        ;
1393 12EC        ;                   ... BIT 7 ...
1394 12EC 906E A         LD      AC0,@APIP
1395 12ED 1103 A         BOC     CCEZ,.+4        ; BRANCH IF NO PREEMPT IN PROGRESS
1396 12EE 3481 A         RCPY    AC1,AC0
1397 12EF 0727 A         SETBIT  D7              ; PREEMPT IN PROGRESS
1398 12F0 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
1399 12F1        ;
1400 12F1        ;                   ... BIT 6 ...
1401 12F1 907B A         LD      AC0,@AVCBT      ; VEHICLE CLASSIFICATION FLAG
1402 12F2 1103 A         BOC     CCEZ,.+4        ; BRANCH IF NO RECENT BUS CLASS.
1403 12F3 3481 A         RCPY    AC1,AC0
1404 12F4 0726 A         SETBIT  D6              ; BUS CLASSIFIED DURING NEAR PAST
1405 12F5 3181 A         RCPY    AC0,AC1         ; STORE TEMPORARY RESULT
1406 12F6        ;
1407 12F6        ;                   ... BITS 5 THRU 0 REQUIRE MANUAL RESET ...
1408 12F6        ;
1409 12F6        ;                   ... BIT 5 ...
1410 12F6 909F A         LD      AC0,@AEERR
1411 12F7 6121 A         AND     AC0,HFFEF       ; IGNORE DETECTOR UNDERCOUNT ERROR
1412 12F8 1508 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
1413 12F9 906C A         LD      AC0,@AXERR
1414 12FA 1506 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
1415 12FB 906A A         LD      AC0,@ABERR
1416 12FC 1504 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
1417 12FD 9090 A         LD      AC0,@ATERR
1418 12FE 1502 A         BOC     CCNZ,ESDW1      ; BRANCH IF ERROR
1419 12FF 9077 A         LD      AC0,@APERR
1420 1300 1103 A         BOC     CCEZ,.+4
1421 1301        ESDW1:
1422 1301 3481 A         RCPY    AC1,AC0
1423 1302 0725 A         SETBIT  D5              ; SOFTWARE DETECTED ERROR
```

```
1424 1303 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1425 1304          ;
1426 1304          ;                   ... BIT 4 ...
1427 1304 909F A          LD      AC0,@AEERR
1428 1305 6114 A          AND     AC0,H0010        ; EXTRACT DETECTOR UNDERCOUNT FLAG
1429 1306 1103 A          BOC     CCEZ,.+4         ; BRANCH IF NO DETECTOR UNDERCOUNT ERROR
1430 1307 3481 A          RCPY    AC1,AC0
1431 1308 0724 A          SETBIT  D4               ; DETECTOR UNDERCOUNT ERROR
1432 1309 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1433 130A          ;
1434 130A          ;                   ... BIT 2 ...
1435 130A 9069 A          LD      AC0,@ABDIN       ; DISCRETE INPUT WORD
1436 130B 0756 A          SKBIT   D6               ; CMU CONFLICT INDICATION
1437 130C 2103 A          JMP     .+4
1438 130D 3481 A          RCPY    AC1,AC0
1439 130E 0722 A          SETBIT  D2               ; CMU CONFLICT
1440 130F 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1441 1310          ;
1442 1310          ;                   ... BIT 1 ...
1443 1310 9069 A          LD      AC0,@ABDIN       ; DISCRETE INPUT WORD
1444 1311 075F A          SKBIT   D15              ; SKIP IF EXTERNAL DISCONNECT
1445 1312 2103 A          JMP     .+4
1446 1313 3481 A          RCPY    AC1,AC0
1447 1314 0721 A          SETBIT  D1               ; EXTERNAL DISCONNECT
1448 1315 3181 A          RCPY    AC0,AC1          ; STORE TEMPORARY RESULT
1449 1316          ;
1450 1316 B46B A          ST      AC1,@AXBUF       ; STORE STATUS DISPLAY WORD
1451 1317          ;
1452 1317 0200 A          RTS     D0               ; RETURN FROM ESDW
1453 1318          ;
1454 1318          ;                   ... LOCAL DATA ...
1455 1318 F800 A HF800:   .WORD   X'F800           ; BITS 10 TO 0 RESET
1456 1319 FFEF A HFFEF:   .WORD   X'FFEF           ; BIT 4 RESET
1457 131A 0010 A H0010:   .WORD   X'0010           ; BIT 4 SET
1458 131B          ;
1459 131B          ;
```

```
1462 131B                .LIST   LES
1463 131B        ;
1464 131B        ;                  ... END OF CYCLE ROUTINE ...
1465 131B        ;
1466 131B        EEOC:
1467 131B 4C01 A         LI      AC0,C1
1468 131C B086 A         ST      AC0,@ATFST       ; SET 'WAIT FOR SYNCH START' FLAG
1469 131D        ;
1470 131D        ;                  ... SUBSEQUENT FNS MAY RESET THIS FLAG ...
1471 131D        ;
1472 131D        ;
1473 131D        ;                  ... PREEMPT FUNCTIONS ...
1474 131D 2903 A         JSR     EPEOC
1475 131E        ;
1476 131E        ;                  ... TIMING PLAN FUNCTIONS ...
1477 131E 2C37 A         JSR     @ATNCY
1478 131F        ;
1479 131F        ;                  ... EXECUTIVE TIMING FUNCTIONS ...
1480 131F 2913 A         JSR     ETEOC
1481 1320        ;
1482 1320 0200 A         RTS     D0               ; RETURN FROM EEOC
1483 1321        ;
1484 1321        ;                  ... END OF CYCLE PREEMPT FUNCTIONS ...
1485 1321        EPEOC:
1486 1321 9C74 A         LD      AC0,@APFREQ      ; TEST FOR PREEMPT MOD OF THE PAST CYCLE
1487 1322 1501 A         BOC     CCNZ,.+2         ; BRANCH IF PREEMPT MOD
1488 1323 2102 A         JMP     .+3
1489 1324 4C00 A         LI      AC0,C0
1490 1325 B086 A         ST      AC0,@ATFST       ; RESET 'WAIT FOR SYNC START' FLAG
1491 1326 4C00 A         LI      AC0,C0
1492 1327 B074 A         ST      AC0,@APFREQ      ; RESET PREEMPT MOD FLAG
1493 1328        ;
1494 1328                                         ; CLEAR PREEMPT MONITOR INDEX FLAGS
1495 1328 8873 A         LD      AC2,APMIF
1496 1329 8200 A         LD      AC0,D0(AC2)
1497 132A 60F0 A         AND     AC0,H7FFF        ; RESET BIT 15
1498 132B A200 A         ST      AC0,D0(AC2)
1499 132C 8201 A         LD      AC0,D1(AC2)
1500 132D 60F0 A         AND     AC0,H7FFF        ; RESET BIT 15
1501 132E A201 A         ST      AC0,D1(AC2)
1502 132F 8202 A         LD      AC0,D2(AC2)
1503 1330 60F0 A         AND     AC0,H7FFF        ; RESET BIT 15
1504 1331 A202 A         ST      AC0,D2(AC2)
1505 1332        ;
1506 1332 0200 A         RTS     D0               ; RETURN FROM EPEOC
1507 1333        ;
1508 1333        ;                  ... END OF CYCLE EXECUTIVE TIMING ROUTINE ...
1509 1333        ETEOC:
1510 1333 889B A         LD      AC2,AETCY        ; TIMING PLAN CYCLE TIMER
1511 1334 8600 A         LD      AC1,D0(AC2)
1512 1335 E601 A         SKG     AC1,D1(AC2)      ; SKIP IF GREATER THAN PREVIOUS MAXIMUM
1513 1336 2102 A         JMP     ETEOC1
```

```
1514 1337 A601 A          ST      AC1,D1(AC2)      ; STORE NEW MAXIMUM
1515 1338 2104 A          JMP     ETEOC2
1516 1339 3481 A ETEOC1:  RCPY    AC1,AC0
1517 133A D202 A          SUB     AC0,D2(AC2)
1518 133B 1201 A          DOC     CCZP,ETEOC2      ; BRANCH IF GREATER THAN PREVIOUS MIN.
1519 133C A602 A          ST      AC1,D2(AC2)
1520 133D 4D00 A ETEOC2:  LI      AC1,C0
1521 133E A600 A          ST      AC1,D0(AC2)      ; ZERO CYCLE TIMER
1522 133F        ;
1523 133F 0200 A          RTS     D0               ; RETURN FROM ETEOC
1524 1340        ;
```

```
1340                .LIST   LER
1340        ;
1340        ;                   ... ERROR RECORDING ROUTINE ...
1340        ;
1340        ;                         UPON ENTRY, AC1 = ERROR DATUM
1340        ;                                     AC3 = BASE ADDR OF ERROR BUFFER
1340        ;                                     AC0 USED TO SET ERROR FLAG
1340        ;                         ENTRY POINT IMPLIES ERROR TYPE
1340        ;
1340        ERE0:
1340 A702 A         ST      AC1,D2(AC3)     ; STORE ERROR DATUM
1341 8301 A         LD      AC0,D1(AC3)     ; FETCH ERROR COUNTER
1342 4801 A         AISZ    AC0,C1          ; INCREMENT
1343 2100 A         JMP     .+1
1344 A301 A         ST      AC0,D1(AC3)
1345 8300 A         LD      AC0,D0(AC3)
1346 0720 A         SETBIT  D0              ; SET ERROR FLAG BIT 0
1347 A300 A         ST      AC0,D0(AC3)
1348 0200 A         RTS     D0              ; RETURN FROM ERE0
1349        ;
1349        ERE1:
1349 A704 A         ST      AC1,D4(AC3)     ; STORE ERROR DATUM
134A 8303 A         LD      AC0,D3(AC3)     ; FETCH ERROR COUNTER
134B 4801 A         AISZ    AC0,C1          ; INCREMENT
134C 2100 A         JMP     .+1
134D A303 A         ST      AC0,D3(AC3)
134E 8300 A         LD      AC0,D0(AC3)
134F 0721 A         SETBIT  D1              ; SET ERROR FLAG BIT 1
1350 A300 A         ST      AC0,D0(AC3)
1351 0200 A         RTS     D0              ; RETURN FROM ERE1
1352        ;
1352        ERE2:
1352 A706 A         ST      AC1,D6(AC3)     ; STORE ERROR DATUM
1353 8305 A         LD      AC0,D5(AC3)     ; FETCH ERROR COUNTER
1354 4801 A         AISZ    AC0,C1          ; INCREMENT
1355 2100 A         JMP     .+1
1356 A305 A         ST      AC0,D5(AC3)
1357 8300 A         LD      AC0,D0(AC3)
1358 0722 A         SETBIT  D2              ; SET ERROR FLAG BIT 2
1359 A300 A         ST      AC0,D0(AC3)
135A 0200 A         RTS     D0              ; RETURN FROM ERE2
135B        ;
135B        ERE3:
135B A708 A         ST      AC1,D8(AC3)     ; STORE ERROR DATUM
135C 8307 A         LD      AC0,D7(AC3)     ; FETCH ERROR COUNTER
135D 4801 A         AISZ    AC0,C1          ; INCREMENT
135E 2100 A         JMP     .+1
135F A307 A         ST      AC0,D7(AC3)
1360 8300 A         LD      AC0,D0(AC3)
1361 0723 A         SETBIT  D3              ; SET ERROR FLAG BIT 3
1362 A300 A         ST      AC0,D0(AC3)
1363 0200 A         RTS     D0              ; RETURN FROM ERE3
```

```
1364          ;
1364          ERE4:
1364 A70A A          ST      AC1,D10(AC3)    ; STORE ERROR DATUM
1365 8309 A          LD      AC0,D9(AC3)     ; FETCH ERROR COUNTER
1366 4801 A          AISZ    AC0,C1          ; INCREMENT
1367 2100 A          JMP     .+1
1368 A309 A          ST      AC0,D9(AC3)
1369 8300 A          LD      AC0,D0(AC3)
136A 0724 A          SETBIT  D4              ; SET ERROR FLAG BIT 4
136B A300 A          ST      AC0,D0(AC3)
136C 0200 A          RTS     D0              ; RETURN FROM ERE4
136D          ;
136D          ERE5:
136D A70C A          ST      AC1,D12(AC3)    ; STORE ERROR DATUM
136E 830B A          LD      AC0,D11(AC3)    ; FETCH ERROR COUNTER
136F 4801 A          AISZ    AC0,C1          ; INCREMENT
1370 2100 A          JMP     .+1
1371 A30B A          ST      AC0,D11(AC3)
1372 8300 A          LD      AC0,D0(AC3)
1373 0725 A          SETBIT  D5              ; SET ERROR FLAG BIT 5
1374 A300 A          ST      AC0,D0(AC3)
1375 0200 A          RTS     D0              ; RETURN FROM ERE5
1376          ;
1376          ERE6:
1376 A70E A          ST      AC1,D14(AC3)    ; STORE ERROR DATUM
1377 830D A          LD      AC0,D13(AC3)    ; FETCH ERROR COUNTER
1378 4801 A          AISZ    AC0,C1          ; INCREMENT
1379 2100 A          JMP     .+1
137A A30D A          ST      AC0,D13(AC3)
137B 8300 A          LD      AC0,D0(AC3)
137C 0726 A          SETBIT  D6              ; SET ERROR FLAG BIT 6
137D A300 A          ST      AC0,D0(AC3)
137E 0200 A          RTS     D0              ; RETURN FROM ERE6
137F          ;
137F          ERE7:
137F A710 A          ST      AC1,D16(AC3)    ; STORE ERROR DATUM
1380 830F A          LD      AC0,D15(AC3)    ; FETCH ERROR COUNTER
1381 4801 A          AISZ    AC0,C1          ; INCREMENT
1382 2100 A          JMP     .+1
1383 A30F A          ST      AC0,D15(AC3)
1384 8300 A          LD      AC0,D0(AC3)
1385 0727 A          SETBIT  D7              ; SET ERROR FLAG BIT 7
1386 A300 A          ST      AC0,D0(AC3)
1387 0200 A          RTS     D0              ; RETURN FROM ERE7
1388          ;
1388          ERE8:
1388 A712 A          ST      AC1,D18(AC3)    ; STORE ERROR DATUM
1389 8311 A          LD      AC0,D17(AC3)    ; FETCH ERROR COUNTER
138A 4801 A          AISZ    AC0,C1          ; INCREMENT
138B 2100 A          JMP     .+1
138C A311 A          ST      AC0,D17(AC3)
138D 8300 A          LD      AC0,D0(AC3)
```

```
1631 138E 0728 A        SETBIT  D8              ; SET ERROR FLAG BIT 8
1632 138F A300 A        ST      AC0,D0(AC3)
1633 1390 0200 A        RTS     D0              ; RETURN FROM ERE8
1634 1391        ;
1635 1391        ERE9:
1636 1391 A714 A        ST      AC1,D20(AC3)    ; STORE ERROR DATUM
1637 1392 8313 A        LD      AC0,D19(AC3)    ; FETCH ERROR COUNTER
1638 1393 4801 A        AISZ    AC0,C1          ; INCREMENT
1639 1394 2100 A        JMP     .+1
1640 1395 A313 A        ST      AC0,D19(AC3)
1641 1396 8300 A        LD      AC0,D0(AC3)
1642 1397 0729 A        SETBIT  D9              ; SET ERROR FLAG BIT 9
1643 1398 A300 A        ST      AC0,D0(AC3)
1644 1399 0200 A        RTS     D0              ; RETURN FROM ERE9
1645 139A        ;
1646 139A        ERE10:
1647 139A A716 A        ST      AC1,D22(AC3)    ; STORE ERROR DATUM
1648 139B 8315 A        LD      AC0,D21(AC3)    ; FETCH ERROR COUNTER
1649 139C 4801 A        AISZ    AC0,C1          ; INCREMENT
1650 139D 2100 A        JMP     .+1
1651 139E A315 A        ST      AC0,D21(AC3)
1652 139F 8300 A        LD      AC0,D0(AC3)
1653 13A0 072A A        SETBIT  D10             ; SET ERROR FLAG BIT 10
1654 13A1 A300 A        ST      AC0,D0(AC3)
1655 13A2 0200 A        RTS     D0              ; RETURN FROM ERE10
1656 13A3        ;
1657 13A3        ERE11:
1658 13A3 A718 A        ST      AC1,D24(AC3)    ; STORE ERROR DATUM
1659 13A4 8317 A        LD      AC0,D23(AC3)    ; FETCH ERROR COUNTER
1660 13A5 4801 A        AISZ    AC0,C1          ; INCREMENT
1661 13A6 2100 A        JMP     .+1
1662 13A7 A317 A        ST      AC0,D23(AC3)
1663 13A8 8300 A        LD      AC0,D0(AC3)
1664 13A9 072B A        SETBIT  D11             ; SET ERROR FLAG BIT 11
1665 13AA A300 A        ST      AC0,D0(AC3)
1666 13AB 0200 A        RTS     D0              ; RETURN FROM ERE11
1667 13AC        ;
1668 13AC        ERE12:
1669 13AC A71A A        ST      AC1,D26(AC3)    ; STORE ERROR DATUM
1670 13AD 8319 A        LD      AC0,D25(AC3)    ; FETCH ERROR COUNTER
1671 13AE 4801 A        AISZ    AC0,C1          ; INCREMENT
1672 13AF 2100 A        JMP     .+1
1673 13B0 A319 A        ST      AC0,D25(AC3)
1674 13B1 8300 A        LD      AC0,D0(AC3)
1675 13B2 072C A        SETBIT  D12             ; SET ERROR FLAG BIT 12
1676 13B3 A300 A        ST      AC0,D0(AC3)
1677 13B4 0200 A        RTS     D0              ; RETURN FROM ERE12
1678 13B5        ;
```

```
1679 13B5                .PAGE
1680 13B5                .ASECT
1681 13B5 1400 A         .=X'1400
1682 1400        ;
```

```
1400                    .LIST   LPP
1400          ;
1400          ;                    ... PROGRAMMABLE PANEL ROUTINE ...
1400          ;
1400          ;              (EXECUTED DURING RESTART INITIALIZATION ONLY)
1400          ;
1400          EPAN:
1400          ;
1400          ;                    ... INPUT PROG. PANEL DATA ...
1400 2C3E A          JSR     @ABPAN
1401          ;
1401          ;                    ... CLEAR ERROR FLAGS ...
1401 4C00 A          LI      AC0,C0
1402 B0CD A          ST      AC0,@AEP1E        ; PLAN 1
1403 B0CE A          ST      AC0,@AEP2E        ; PLAN 2
1404 B0CF A          ST      AC0,@AEP3E        ; PLAN 3
1405 B0D0 A          ST      AC0,@AEPDVE       ; DATA VALIDITY
1406          ;
1406          ;                    ... TURN OFF ERROR LIGHTS ...
1406 8CEA A          LD      AC3,APP1U         ; PERIPHERAL DEVICE ADDRESS PLAN 1
1407 2C57 A          JSR     @AXLOF1
1408          ;
1408 8CEB A          LD      AC3,APP2U         ; PERIPHERAL DEVICE ADDRESS PLAN 2
1409 2C57 A          JSR     @AXLOF1
140A          ;
140A 8CEC A          LD      AC3,APP3U         ; PERIPHERAL DEVICE ADDRESS PLAN 3
140B 2C57 A          JSR     @AXLOF1
140C          ;
140C 8CE9 A          LD      AC3,APPDU         ; DEVICE ADDRESS DATA VALIDITY LIGHT
140D 2C59 A          JSR     @AXLOF2
140E          ;
140E          ;                    ... COMPUTE SUM OF FIXED INTERVALS IN PLAN 1 ...
140E 888B A          LD      AC2,ATPM1
140F 2951 A          JSR     EPFI
1410 8CC5 A          LD      AC3,AEPFT
1411 A700 A          ST      AC1,D0(AC3)       ; STORE SUM
1412          ;
1412          ;                    ... COMPUTE SUM OF FIXED INTERVALS IN PLAN 2 ...
1412 888C A          LD      AC2,ATPM2
1413 294D A          JSR     EPFI
1414 8CC5 A          LD      AC3,AEPFT
1415 A701 A          ST      AC1,D1(AC3)       ; STORE SUM
1416          ;
1416          ;                    ... COMPUTE SUM OF FIXED INTERVALS IN PLAN 3 ...
1416 888D A          LD      AC2,ATPM3
1417 2949 A          JSR     EPFI
1418 8CC5 A          LD      AC3,AEPFT
1419 A702 A          ST      AC1,D2(AC3)       ; STORE SUM
141A          ;
141A          ;                    ... TEST ADJUSTABLE INTERVALS FOR 0 DURATION ...
141A 888B A          LD      AC2,ATPM1
141B 9493 A          LD      AC1,@ATAI1        ; RELATIVE INDEX OF ADJ INT 1
```

```
141C 3600 A          RADD   AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 1
141D 8200 A          LD     AC0,D0(AC2)      ; FETCH DURATION
141E 1107 A          BOC    CCEZ,EPAN1       ; BRANCH IF ZERO, NO ERROR
141F 8CD1 A          LD     AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
1420          ;
1420          ;                 ... RECORD ERROR IN MASTER TIMING PLAN ...
1420          EPANE:
1420 3981 A          RCPY   AC2,AC1          ; SAVE ADDRESS AS ERROR DATUM
1421 2C49 A          JSR    @AERE0
1422 2C3A A EPANE1:  JSR    @AXWDT           ; ISSUE RESET PULSE TO WATCHDOG TIMER
1423          ;
1423 2C2F A          JSR    @AEADC           ; FATAL ERROR, DISCONNECT, NO RETURN
1424          ;
1424 2100 A          JMP    .+1              ; THIS INST SHOULD NEVER BE REACHED
1425 21FE A          JMP    .-1
1426          ;
1426          EPAN1:
1426 888B A          LD     AC2,ATPM1
1427 9494 A          LD     AC1,@ATAI2       ; RELATIVE INDEX OF ADJ INT 2
1428 3600 A          RADD   AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 2
1429 8200 A          LD     AC0,D0(AC2)      ; FETCH DURATION
142A 1102 A          BOC    CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
142B 8CD1 A          LD     AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
142C 21F3 A          JMP    EPANE
142D          ;
142D 888C A          LD     AC2,ATPM2
142E 9493 A          LD     AC1,@ATAI1       ; RELATIVE INDEX OF ADJ INT 1
142F 3600 A          RADD   AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 1
1430 8200 A          LD     AC0,D0(AC2)      ; FETCH DURATION
1431 1102 A          BOC    CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1432 8CD2 A          LD     AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1433 21EC A          JMP    EPANE
1434          ;
1434 888C A          LD     AC2,ATPM2
1435 9494 A          LD     AC1,@ATAI2       ; RELATIVE INDEX OF ADJ INT 2
1436 3600 A          RADD   AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 2
1437 8200 A          LD     AC0,D0(AC2)      ; FETCH DURATION
1438 1102 A          BOC    CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1439 8CD2 A          LD     AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
143A 21E5 A          JMP    EPANE
143B          ;
143B 888D A          LD     AC2,ATPM3
143C 9493 A          LD     AC1,@ATAI1       ; RELATIVE INDEX OF ADJ INT 1
143D 3600 A          RADD   AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 1
143E 8200 A          LD     AC0,D0(AC2)      ; FETCH DURATION
143F 1102 A          BOC    CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1440 8CD3 A          LD     AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
1441 21DE A          JMP    EPANE
1442          ;
1442 888D A          LD     AC2,ATPM3
1443 9494 A          LD     AC1,@ATAI2       ; RELATIVE INDEX OF ADJ INT 2
1444 3600 A          RADD   AC1,AC2          ; COMPUTE ADDRESS OF ADJUSTABLE INT 2
```

```
1789 1445 8200 A        LD      AC0,D0(AC2)      ; FETCH DURATION
1790 1446 1102 A        BOC     CCEZ,.+3         ; BRANCH IF ZERO, NO ERROR
1791 1447 8CD3 A        LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
1792 1448 21D7 A        JMP     EPANE
1793 1449        ;
1794 1449        ;                  ... PROCESS PLAN 1 DATA ...
1795 1449 2C41 A        JSR     @AEPP1
1796 144A        ;
1797 144A        ;                  ... PROCESS PLAN 2 DATA ...
1798 144A 2C42 A        JSR     @AEPP2
1799 144B        ;
1800 144B        ;                  ... PROCESS PLAN 3 DATA ...
1801 144B 2C43 A        JSR     @AEPP3
1802 144C        ;
1803 144C        ;                  ... TEST ERROR FLAGS ...
1804 144C 90CD A        LD      AC0,@AEP1E       ; PLAN 1
1805 144D 1102 A        BOC     CCEZ,.+3         ; BRANCH IF NO ERROR
1806 144E 8CEA A        LD      AC3,APP1U
1807 144F 2C56 A        JSR     @AXLON1          ; TURN ON ERROR LIGHT
1808 1450        ;
1809 1450 90CE A        LD      AC0,@AEP2E       ; PLAN 2
1810 1451 1102 A        BOC     CCEZ,.+3         ; BRANCH IF NO ERROR
1811 1452 8CEB A        LD      AC3,APP2U
1812 1453 2C56 A        JSR     @AXLON1          ; TURN ON ERROR LIGHT
1813 1454        ;
1814 1454 90CF A        LD      AC0,@AEP3E       ; PLAN 3
1815 1455 1102 A        BOC     CCEZ,.+3         ; BRANCH IF NO ERROR
1816 1456 8CEC A        LD      AC3,APP3U
1817 1457 2C56 A        JSR     @AXLON1          ; TURN ON ERROR LIGHT
1818 1458        ;
1819 1458 90D0 A        LD      AC0,@AEPDVE      ; DATA VALIDITY
1820 1459 1103 A        BOC     CCEZ,.+4         ; BRANCH IF NO ERROR
1821 145A 8CE9 A        LD      AC3,APPDU
1822 145B 2C58 A        JSR     @AXLON2          ; TURN ON ERROR LIGHT
1823 145C 21C5 A        JMP     EPANE1           ; DISCONNECT
1824 145D        ;
1825 145D        ;                  ... UPDATE DATA BASE ...
1826 145D 2C44 A        JSR     @AEPUP
1827 145E        ;
1828 145E 8CE9 A        LD      AC3,APPDU
1829 145F 2C57 A        JSR     @AXLOF1          ; ASSURE PENDING INPUT LIGHT IS OFF
1830 1460        ;
1831 1460 0200 A        RTS     D0               ; RETURN FROM EPAN
1832 1461        ;
1833 1461        ;                  ... COMPUTE SUM OF INTERVALS IN MASTER PLAN ...
1834 1461
1835 1461        ;                        AC2 = BASE ADDRESS OF MASTER TIMING PLAN
1836 1461        ;
1837 1461        ;                        AC1 RETURNS SUM OF INTERVAL DURATIONS
1838 1461        ;
1839 1461        EPFI:
1840 1461 4D00 A        LI      AC1,C0           ; CLEAR SUM
```

```
1841 1462 8200 A EPFI1:  LD      AC0,D0(AC2)     ; FETCH INTERVAL DURATION
1842 1463 1102 A         BOC     CCEZ,.+3        ; BRANCH IF ZERO
1843 1464 1804 A         BOC     CCZN,EPFIR      ; BRANCH IF NEGATIVE, END OF PLAN
1844 1465 3100 A         RADD    AC0,AC1         ; ADD DURATION TO SUM
1845 1466 4A01 A         AISZ    AC2,C1          ; INCREMENT ADDRESS
1846 1467 2100 A         JMP     .+1
1847 1468 21F9 A         JMP     EPFI1           ; CONTINUE LOOP
1848 1469        ;
1849 1469 0200 A EPFIR:  RTS     D0              ; RETURN FROM EPFI
1850 146A        ;
1851 146A        ;                    ... LOCAL DATA ...
1852 146A 00C8 A EPCY:   .WORD   200             ; MAXIMUM CYCLE LENGTH (SECONDS)
1853 146B 00C8 A EPIN1:  .WORD   200             ; LIMIT ON MAXIMUM EXTENSION (SECONDS)
1854 146C 000A A E10:    .WORD   10              ; CONSTANT FOR DIVISION
1855 146D 0064 A E100:   .WORD   100             ; PERCENTAGE CONVERSION CONSTANT
1856 146E        ;
```

```
1859 146E                    .LIST   LPP
1860 146E          ;
1861 146E          ;                 ... PROG PANEL MONITOR ROUTINE ...
1862 146E          ;
1863 146E          ;                     (EXECUTED DURING RTC SEQUENCE)
1864 146E          ;
1865 146E          EPPM:
1866 146E 90BB A          LD      AC0,@ABPPD      ; FETCH PROG PANEL DISCRETES
1867 146F 6153 A          AND     AC0,H0400       ; EXTRACT INPUT REQUEST SWITCH STATE
1868 1470 1502 A          BOC     CCNZ,.+3        ; BRANCH IF INPUT REQUEST ON
1869 1471          ;
1870 1471 B0C6 A          ST      AC0,@AEPUF      ; (AC0) = 0, ASSURE UPDATE FLAG RESET
1871 1472 2105 A          JMP     EPPM1
1872 1473          ;
1873 1473 90C8 A          LD      AC0,@AEPIR      ; FETCH PREVIOUS STATE OF INPUT REQ
1874 1474 1503 A          BOC     CCNZ,.+4        ; BRANCH IF SWITCH WAS ON
1875 1475          ;
1876 1475 4CFF A          LI      AC0,CM1
1877 1476 B0C7 A          ST      AC0,@AEPCF      ; SET FLAG FOR INPUT OF THUMBWHEELS
1878 1477 814B A          LD      AC0,H0400       ; RESTORE SWITCH DATUM
1879 1478          ;
1880 1478 B0C8 A EPPM1:   ST      AC0,@AEPIR      ; SAVE STATE OF INPUT REQUEST
1881 1479          ;
1882 1479 90C7 A          LD      AC0,@AEPCF
1883 147A 1147 A          BOC     CCEZ,EPPMR      ; BRANCH IF NOT INPUT PROCESSING SEQ
1884 147B          ;
1885 147B 88C7 A          LD      AC2,AEPCF       ; ADDRESS OF COMPUTATION FLAG
1886 147C          ;
1887 147C 4DFF A          LI      AC1,CM1
1888 147D F600 A          SKNE    AC1,(AC2)       ; SKIP IF NOT THUMBWHEEL INPUT
1889 147E 2101 A          JMP     .+2
1890 147F 210B A          JMP     EPPM2
1891 1480          ;
1892 1480          ;                 ... CLEAR ERROR FLAGS ...
1893 1480 4C00 A          LI      AC0,C0
1894 1481 B0CD A          ST      AC0,@AEP1E      ; PLAN 1
1895 1482 B0CE A          ST      AC0,@AEP2E      ; PLAN 2
1896 1483 B0CF A          ST      AC0,@AEP3E      ; PLAN 3
1897 1484 B0D0 A          ST      AC0,@AEPDVE     ; DATA VALIDITY
1898 1485          ;
1899 1485          ;                 ... TURN OFF ERROR LIGHT ...
1900 1485          ;
1901 1485 8CE9 A          LD      AC3,APPDU       ; DEVICE ADDRESS DATA VALIDITY LIGHT
1902 1486 2C59 A          JSR     @AXLOF2
1903 1487          ;
1904 1487 2C3E A          JSR     @ABPAN          ; INPUT THUMBWHEEL DATA
1905 1488          ;
1906 1488 4D01 A          LI      AC1,C1
1907 1489 B4C7 A          ST      AC1,@AEPCF      ; SET FLAG FOR PLAN 1 PROCESSING
1908 148A 2137 A          JMP     EPPMR
1909 148B          ;
1910 148B 4D01 A EPPM2:   LI      AC1,C1
```

```
1911 148C F600 A          SKNE   AC1,(AC2)        ; SKIP IF NOT PLAN 1 PROCESSING
1912 148D 2101 A          JMP    .+2
1913 148E 210A A          JMP    EPPM3
1914 148F          ;
1915 148F          ;                 ... TURN OFF ERROR LIGHT ...
1916 148F 8CEA A          LD     AC3,APP1U        ; PERIPHERAL DEVICE ADDRESS PLAN 1
1917 1490 2C57 A          JSR    @AXLOF1
1918 1491          ;
1919 1491 2C41 A          JSR    @AEPP1           ; PLAN 1 PROCESSING
1920 1492          ;
1921 1492          ;                 ... TEST ERROR FLAG ...
1922 1492 90CD A          LD     AC0,@AEP1E       ; PLAN 1
1923 1493 1102 A          BOC    CCEZ,.+3         ; BRANCH IF NO ERROR
1924 1494 8CEA A          LD     AC3,APP1U
1925 1495 2C56 A          JSR    @AXLON1          ; TURN ON ERROR LIGHT
1926 1496          ;
1927 1496 4D02 A          LI     AC1,C2
1928 1497 B4C7 A          ST     AC1,@AEPCF       ; SET FLAG FOR PLAN 2 PROCESSING
1929 1498 2129 A          JMP    EPPMR
1930 1499          ;
1931 1499 4D02 A EPPM3:   LI     AC1,C2
1932 149A F600 A          SKNE   AC1,(AC2)        ; SKIP IF NOT PLAN 2 PROCESSING
1933 149B 2101 A          JMP    .+2
1934 149C 210A A          JMP    EPPM4
1935 149D          ;
1936 149D          ;                 ... TURN OFF ERROR LIGHT ...
1937 149D          ;
1938 149D 8CEB A          LD     AC3,APP2U        ; PERIPHERAL DEVICE ADDRESS PLAN 2
1939 149E 2C57 A          JSR    @AXLOF1
1940 149F          ;
1941 149F 2C42 A          JSR    @AEPP2           ; PLAN 2 PROCESSING
1942 14A0          ;
1943 14A0          ;                 ... TEST ERROR FLAG ...
1944 14A0 90CE A          LD     AC0,@AEP2E       ; PLAN 2
1945 14A1 1102 A          BOC    CCEZ,.+3         ; BRANCH IF NO ERROR
1946 14A2 8CEB A          LD     AC3,APP2U
1947 14A3 2C56 A          JSR    @AXLON1          ; TURN ON ERROR LIGHT
1948 14A4          ;
1949 14A4 4D03 A          LI     AC1,C3
1950 14A5 B4C7 A          ST     AC1,@AEPCF       ; SET FLAG FOR PLAN 3 PROCESSING
1951 14A6 211B A          JMP    EPPMR
1952 14A7          ;
1953 14A7 4D03 A EPPM4:   LI     AC1,C3
1954 14A8 F600 A          SKNE   AC1,(AC2)        ; SKIP IF NOT PLAN 3 PROCESSING
1955 14A9 2101 A          JMP    .+2
1956 14AA 210A A          JMP    EPPM5
1957 14AB          ;
1958 14AB          ;                 ... TURN OFF ERROR LIGHT ...
1959 14AB 8CEC A          LD     AC3,APP3U        ; PERIPHERAL DEVICE ADDRESS PLAN 3
1960 14AC 2C57 A          JSR    @AXLOF1
1961 14AD          ;
1962 14AD 2C43 A          JSR    @AEPP3           ; PLAN 3 PROCESSING
```

```
14AE          ;
14AE          ;                    ... TEST ERROR FLAG ...
14AE 90CF A          LD      AC0,@AEP3E      ; PLAN 3
14AF 1102 A          BOC     CCEZ,.+3        ; BRANCH IF NO ERROR
14B0 8CEC A          LD      AC3,APP3U
14B1 2C56 A          JSR     @AXLON1         ; TURN ON ERROR LIGHT
14B2          ;
14B2 4D04 A          LI      AC1,C4
14B3 B4C7 A          ST      AC1,@AEPCF      ; SET FLAG FOR COMPLETION OF INPUT PROC
14B4 210D A          JMP     EPPMR
14B5          ;
14B5 4D04 A EPPM5:   LI      AC1,C4
14B6 F600 A          SKNE    AC1,(AC2)       ; SKIP IF NOT COMPLETION OF INPUT PROC
14B7 2101 A          JMP     .+2
14B8 2109 A          JMP     EPPMR
14B9          ;
14B9          ;                    ... TEST ERROR FLAG ...
14B9 90D0 A          LD      AC0,@AEPDVE     ; DATA VALIDITY
14BA 1103 A          BOC     CCEZ,.+4        ; BRANCH IF NO ERROR
14BB 8CE9 A          LD      AC3,APPDU
14BC 2C58 A          JSR     @AXLON2         ; TURN ON ERROR LIGHT
14BD 2102 A          JMP     .+3             ; DATA VALIDITY ERROR, DO NOT UPDATE
14BE          ;
14BE 4D01 A          LI      AC1,C1
14BF B4C6 A          ST      AC1,@AEPUF      ; SET FLAG FOR DATA BASE UPDATE ENABLE
14C0          ;
14C0 4D00 A          LI      AC1,C0
14C1 B4C7 A          ST      AC1,@AEPCF      ; RESET COMPUTE FLAG
14C2          ;
14C2 0200 A EPPMR:   RTS     DO              ; RETURN FROM EPPM
14C3          ;
14C3          ;                    ... LOCAL DATA ...
14C3 0400 A H0400:   .WORD   X'0400          ; BIT 10 SET
14C4          ;
14C4          ;
```

```
14C4                    .LIST   LPP
14C4            ;
14C4            ;                   ... PROCESS PROG. PANEL DATA FOR PLAN 1 ...
14C4            EPP1:
14C4            ;
14C4            ;                   ... PROCESS CYCLE LENGTH ...
14C4 88BC A             LD      AC2,AEPD1
14C5 8200 A             LD      AC0,D0(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
14C6 2C46 A             JSR     @AEPBI3         ; CONVERTED VALUE RETURNED IN AC1
14C7 212B A             JMP     EPP10           ; DATA VALIDITY ERROR
14C8 8CBF A             LD      AC3,AEPD1
14C9 A700 A             ST      AC1,D0(AC3)     ; SAVE CONVERTED INTEGER VALUE (SECONDS)
14CA 88C9 A             LD      AC2,AEPCY       ; MAXIMUM CYCLE LENGTH TEST (SECONDS)
14CB E600 A             SKG     AC1,D0(AC2)     ; SKIP IF .GT. MAXIMUM
14CC 2107 A             JMP     .+8
14CD            ;
14CD            ;                   ... RECORD CYCLE LENGTH INPUT ERROR ...
14CD 8CD1 A             LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
14CE 2C51 A             JSR     @AERE8          ; AC1 = ERROR DATUM
14CF 4C01 A             LI      AC0,C1
14D0 B0CD A             ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
14D1 94C9 A             LD      AC1,@AEPCY      ; SET CYCLE LENGTH TO MAXIMUM (SECONDS)
14D2 8CBF A             LD      AC3,AEPD1
14D3 A700 A             ST      AC1,D0(AC3)     ; SAVE MAXIMUM CYCLE LENGTH (SECONDS)
14D4            ;
14D4 2C47 A             JSR     @AEM10          ; CONVERT TO 0.1 SECOND UNITS IN AC1
14D5 8CC2 A             LD      AC3,AEPCD1
14D6 A703 A             ST      AC1,D3(AC3)     ; SAVE CYCLE LENGTH (0.1 SEC)
14D7            ;
14D7 88C5 A             LD      AC2,AEPFT       ; MINIMUM CYCLE LENGTH TEST (0.1 SEC)
14D8 8200 A             LD      AC0,D0(AC2)     ; SUM OF FIXED INTERVALS
14D9 5001 A             CAI     AC0,C1          ; NEGATE
14DA 3400 A             PADD    AC1,AC0         ; AC0 = ADJ. TIME IF NON-NEG.
14DB 1B02 A             BOC     CCZN,.+3        ; BRANCH IF CYCLE .LE. MINIMUM
14DC A301 A             ST      AC0,D1(AC3)     ; SAVE ADJUSTABLE TIME IN ADJ. INT. 2
14DD 2109 A             JMP     .+10
14DE            ;
14DE            ;                   ... CYCLE LENGTH INPUT ERROR (TOO SHORT) ...
14DE 4C00 A             LI      AC0,C0          ; NO ADJUSTABLE TIME
14DF A301 A             ST      AC0,D1(AC3)     ; SAVE 0 IN ADJUSTABLE INTERVAL 2
14E0            ;
14E0 8200 A             LD      AC0,D0(AC2)     ; RESTORE MIN. SUM OF FIXED INTERVALS
14E1 8CC2 A             LD      AC3,AEPCD1
14E2 A303 A             ST      AC0,D3(AC3)     ; STORE MIN CYCLE (0.1 SEC)
14E3 8CD1 A             LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
14E4 2C51 A             JSR     @AERE8          ; AC1 = ERROR DATUM
14E5 4C01 A             LI      AC0,C1
14E6 B0CD A             ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
14E7            ;
14E7            ;                   ... COMPUTE MAX DWELL TIME FOR SYNCH (24% OF CYCLE) ...
14E7 8CBF A             LD      AC3,AEPD1
14E8 8700 A             LD      AC1,D0(AC3)     ; CYCLE LENGTH (SECONDS)
```

```
14E9 2C48 A          JSR     @AEM24          ; MULTIPLY BY 24
14EA 4C00 A          LI      AC0,C0          ; PREPARE FOR DIVIDE
14EB 88CC A          LD      AC2,AE10        ; *10/100 TO GET 0.1 SEC UNITS
14EC 0690 A          DIV     D0(AC2)
14ED 0000 A
14EE 8CC2 A          LD      AC3,AEPCD1
14EF A708 A          ST      AC1,D8(AC3)     ; MAX DWELL 24% OF CYCLE (0.1 SEC)
14F0          ;
14F0          ;                  ... COMPUTE START PULSE DURATION (3% OF CYCLE) ...
14F0 5DFD A          SHR     AC1,D3          ; DIVIDE BY 8
14F1 A707 A          ST      AC1,D7(AC3)     ; START PULSE DURATION 3% OF CYCLE
14F2 2105 A          JMP     .+6
14F3          ;
14F3          ;                  ... RECORD CYCLE LENGTH VALIDITY ERROR ...
14F3          EPP10:
14F3 8CD1 A          LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
14F4 2C4A A          JSR     @AERE1          ; AC1 = ERROR DATUM
14F5 4C01 A          LI      AC0,C1
14F6 B0CD A          ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
14F7 B0D0 A          ST      AC0,@AEPDVE     ; SET ERROR FLAG FOR DATA VALIDITY
14F8          ;
14F8          ;                  ... PROCESS OFFSET 1 ...
14F8 88BC A          LD      AC2,ABPD1
14F9 8201 A          LD      AC0,D1(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
14FA 2C46 A          JSR     @AEPBI3         ; CONVERT DATUM TO INTEGER IN AC1
14FB 2110 A          JMP     EPP11           ; DATA VALIDITY ERROR
14FC 8CBF A          LD      AC3,AEPD1
14FD A701 A          ST      AC1,D1(AC3)     ; SAVE CONVERTED VALUE (SECONDS)
14FE E700 A          SKG     AC1,D0(AC3)     ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
14FF 2106 A          JMP     .+7
1500          ;
1500          ;                  ... RECORD OFFSET INPUT ERROR ...
1500 8CD1 A          LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
1501 2C52 A          JSR     @AERE9          ; AC1 = ERROR DATUM
1502 4C01 A          LI      AC0,C1
1503 B0CD A          ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
1504 4DF6 A          LI      AC1,CM10        ; SET OUT OF RANGE RESET TO -1 SECOND
1505 2103 A          JMP     .+4
1506          ;
1506          ;                  ... COMPUTE RESET ...
1506 D700 A          SUB     AC1,D0(AC3)     ; OFFSET - CYCLE
1507 5101 A          CAI     AC1,C1          ; NEGATE
1508 2C47 A          JSR     @AEM10          ; MULTIPLY AC1 BY 10
1509 8CC2 A          LD      AC3,AEPCD1
150A A704 A          ST      AC1,D4(AC3)     ; SAVE RESET (0.1 SECONDS)
150B 2105 A          JMP     .+6
150C          ;
150C          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
150C 8CD1 A EPP11:   LD      AC3,AEPEF1      ; BASE ADDRESS OF ERROR BUFFER
150D 2C4B A          JSR     @AERE2          ; AC1 = ERROR DATUM
150E 4C01 A          LI      AC0,C1
150F B0CD A          ST      AC0,@AEP1E      ; SET ERROR FLAG FOR PLAN 1
```

```
2103 1510 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
2104 1511          ;
2105 1511          ;                  ... PROCESS OFFSET 2 ...
2106 1511 88BC A          LD      AC2,ABPD1
2107 1512 8202 A          LD      AC0,D2(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2108 1513 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
2109 1514 2110 A          JMP     EPP12            ; DATA VALIDITY ERROR
2110 1515 8CBF A          LD      AC3,AEPD1
2111 1516 A702 A          ST      AC1,D2(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
2112 1517 E700 A          SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
2113 1518 2106 A          JMP     .+7
2114 1519          ;
2115 1519          ;                  ... RECORD OFFSET INPUT ERROR ...
2116 1519 8CD1 A          LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2117 151A 2C53 A          JSR     @AERE10          ; AC1 = ERROR DATUM
2118 151B 4C01 A          LI      AC0,C1
2119 151C B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2120 151D 4DF6 A          LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
2121 151E 2103 A          JMP     .+4
2122 151F          ;
2123 151F          ;                  ... COMPUTE RESET ...
2124 151F D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
2125 1520 5101 A          CAI     AC1,C1           ; NEGATE
2126 1521 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
2127 1522 8CC2 A          LD      AC3,AEPCD1
2128 1523 A705 A          ST      AC1,D5(AC3)      ; SAVE RESET (0.1 SECONDS)
2129 1524 2105 A          JMP     .+6
2130 1525          ;
2131 1525          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
2132 1525 8CD1 A EPP12:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2133 1526 2C4C A          JSR     @AERE3           ; AC1 = ERROR DATUM
2134 1527 4C01 A          LI      AC0,C1
2135 1528 B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2136 1529 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
2137 152A          ;
2138 152A          ;                  ... PROCESS OFFSET 3 ...
2139 152A 88BC A          LD      AC2,ABPD1
2140 152B 8203 A          LD      AC0,D3(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2141 152C 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
2142 152D 2110 A          JMP     EPP13            ; DATA VALIDITY ERROR
2143 152E 8CBF A          LD      AC3,AEPD1
2144 152F A703 A          ST      AC1,D3(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
2145 1530 E700 A          SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
2146 1531 2106 A          JMP     .+7
2147 1532          ;
2148 1532          ;                  ... RECORD OFFSET INPUT ERROR ...
2149 1532 8CD1 A          LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2150 1533 2C54 A          JSR     @AERE11          ; AC1 = ERROR DATUM
2151 1534 4C01 A          LI      AC0,C1
2152 1535 B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2153 1536 4DF6 A          LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
2154 1537 2103 A          JMP     .+4
```

```
1538          ;
1538          ;                    ... COMPUTE RESET ...
1538 D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
1539 5101 A          CAI     AC1,C1           ; NEGATE
153A 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
153B 8CC2 A          LD      AC3,AEPCD1
153C A706 A          ST      AC1,D6(AC3)      ; SAVE RESET (0.1 SECONDS)
153D 2105 A          JMP     .+6
153E          ;
153E          ;                    ... RECORD OFFSET DATA VALIDITY ERROR ...
153E 8CD1 A EPP13:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
153F 2C4D A          JSR     @AERE4           ; AC1 = ERROR DATUM
1540 4C01 A          LI      AC0,C1
1541 B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
1542 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1543          ;
1543          ;                    ... PROCESS MSG SPLIT % ...
1543 88BC A          LD      AC2,AEPD1
1544 8204 A          LD      AC0,D4(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1545 2C45 A          JSR     @AEPBI2          ; CONVERT DATUM TO INTEGER IN AC1
1546 2114 A          JMP     EPP14            ; DATA VALIDITY ERROR
1547 8CBF A          LD      AC3,AEPD1
1548 A704 A          ST      AC1,D4(AC3)      ; SAVE CONVERTED VALUE (%)
1549          ;
1549 88C2 A          LD      AC2,AEPCD1
154A 8201 A          LD      AC0,D1(AC2)      ; FETCH ADJUSTABLE TIME
154B 1801 A          BOC     CCZN,.+2         ; TEST FOR POSITIVE
154C 2102 A          JMP     .+3
154D          ;
154D 4D00 A          LI      AC1,C0           ; STORE 0 FOR BOTH ADJ. INTS.
154E 2106 A          JMP     .+7
154F          ;
154F 4C00 A          LI      AC0,C0           ; PREPARE FOR MULTIPLY
1550 0680 A          MPY     D1(AC2)          ; MSG(%) * ADJ. TIME (0.1 SEC)
1551 0001 A
1552 88CB A          LD      AC2,AE100        ; DIVIDE BY 100
1553 0690 A          DIV     D0(AC2)
1554 0000 A
1555          ;
1555 88C2 A          LD      AC2,AEPCD1
1556 A600 A          ST      AC1,D0(AC2)      ; STORE ADJ. INTERVAL 1
1557 5101 A          CAI     AC1,C1           ; SUBTRACT ADJ. INT. 1 FROM TOTAL
1558 C601 A          ADD     AC1,D1(AC2)
1559 A601 A          ST      AC1,D1(AC2)      ; STORE ADJ. INT. 2
155A 2105 A          JMP     .+6
155B          ;
155B          ;                    ... RECORD MSG DATA VALIDITY ERROR ...
155B 8CD1 A EPP14:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
155C 2C4E A          JSR     @AERE5           ; AC1 = ERROR DATUM
155D 4C01 A          LI      AC0,C1
155E B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
155F B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
```

```
2205 1560          ;
2206 1560          ;                    ... PROCESS INT1 ...
2207 1560 88BC A          LD      AC2,ABPD1
2208 1561 8205 A          LD      AC0,D5(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2209 1562 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
2210 1563 210F A          JMP     EPP15            ; DATA VALIDITY ERROR
2211 1564 8C8F A          LD      AC3,AEPD1
2212 1565 A705 A          ST      AC1,D5(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
2213 1566 88CA A          LD      AC2,AEPIN1
2214 1567 E600 A          SKG     AC1,D0(AC2)      ; SKIP IF INT1 .GT. LIMIT
2215 1568 2105 A          JMP     .+6
2216 1569          ;
2217 1569          ;                    ... RECORD INT1 INPUT ERROR ...
2218 1569 8CD1 A          LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2219 156A 2C55 A          JSR     @AERE12          ; AC1 = ERROR DATUM
2220 156B 4C01 A          LI      AC0,C1
2221 156C B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2222 156D 94CA A          LD      AC1,@AEPIN1      ; SET MAXIMUM EXTENSION TO LIMIT (SEC)
2223 156E          ;
2224 156E 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
2225 156F 8CC2 A          LD      AC3,AEPCD1
2226 1570 A702 A          ST      AC1,D2(AC3)      ; STORE MAXIMUM EXTENSION (0.1 SEC)
2227 1571 2105 A          JMP     .+6
2228 1572          ;
2229 1572          ;                    ... RECORD INT1 DATA VALIDITY ERROR ...
2230 1572 8CD1 A EPP15:   LD      AC3,AEPEF1       ; BASE ADDRESS OF ERROR BUFFER
2231 1573 2C4F A          JSR     @AERE6           ; AC1 = ERROR DATUM
2232 1574 4C01 A          LI      AC0,C1
2233 1575 B0CD A          ST      AC0,@AEP1E       ; SET ERROR FLAG FOR PLAN 1
2234 1576 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
2235 1577          ;
2236 1577 0200 A          RTS     D0               ; RETURN FROM EPP1
2237 1578          ;
```

```
1578                    .LIST   LPP
1578          ;
1578          ;                    ... PROCESS PROG. PANEL DATA FOR PLAN 2 ...
1578          EPP2:
1578          ;
1578          ;                    ... PROCESS CYCLE LENGTH ...
1578 88BD A           LD      AC2,AEPD2
1579 8200 A           LD      AC0,D0(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
157A 2C46 A           JSR     @AEPBI3          ; CONVERTED VALUE RETURNED IN AC1
157B 212B A           JMP     EPP20            ; DATA VALIDITY ERROR
157C 8CC0 A           LD      AC3,AEPD2
157D A700 A           ST      AC1,D0(AC3)      ; SAVE CONVERTED INTEGER VALUE (SECONDS)
157E 88C9 A           LD      AC2,AEPCY        ; MAXIMUM CYCLE LENGTH TEST (SECONDS)
157F E600 A           SKG     AC1,D0(AC2)      ; SKIP IF .GT. MAXIMUM
1580 2107 A           JMP     .+8
1581          ;
1581          ;                    ... RECORD CYCLE LENGTH INPUT ERROR ...
1581 8CD2 A           LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1582 2C51 A           JSR     @AERE8           ; AC1 = ERROR DATUM
1583 4C01 A           LI      AC0,C1
1584 B0CE A           ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
1585 94C9 A           LD      AC1,@AEPCY       ; SET CYCLE LENGTH TO MAXIMUM (SECONDS)
1586 8CC0 A           LD      AC3,AEPD2
1587 A700 A           ST      AC1,D0(AC3)      ; SAVE MAXIMUM CYCLE LENGTH (SECONDS)
1588          ;
1588 2C47 A           JSR     @AEM10           ; CONVERT TO 0.1 SECOND UNITS IN AC1
1589 8CC3 A           LD      AC3,AEPCD2
158A A703 A           ST      AC1,D3(AC3)      ; SAVE CYCLE LENGTH (0.1 SEC)
158B          ;
158B 88C5 A           LD      AC2,AEPFT        ; MINIMUM CYCLE LENGTH TEST (0.1 SEC)
158C 8201 A           LD      AC0,D1(AC2)      ; SUM OF FIXED INTERVALS
158D 5001 A           CAI     AC0,C1           ; NEGATE
158E 3400 A           RADD    AC1,AC0          ; AC0 = ADJ. TIME IF NON-NEG.
158F 1802 A           BOC     CCZN,.+3         ; BRANCH IF CYCLE .LE. MINIMUM
1590 A301 A           ST      AC0,D1(AC3)      ; SAVE ADJUSTABLE TIME IN ADJ. INT. 2
1591 2109 A           JMP     .+10
1592          ;
1592          ;                    ... CYCLE LENGTH INPUT ERROR (TOO SHORT) ...
1592 4C00 A           LI      AC0,C0           ; NO ADJUSTABLE TIME
1593 A301 A           ST      AC0,D1(AC3)      ; SAVE 0 IN ADJUSTABLE INTERVAL 2
1594          ;
1594 8200 A           LD      AC0,D0(AC2)      ; RESTORE MIN. SUM OF FIXED INTERVALS
1595 8CC3 A           LD      AC3,AEPCD2
1596 A303 A           ST      AC0,D3(AC3)      ; STORE MIN CYCLE (0.1 SEC)
1597 8CD2 A           LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1598 2C51 A           JSR     @AERE8           ; AC1 = ERROR DATUM
1599 4C01 A           LI      AC0,C1
159A B0CE A           ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
159B          ;
159B          ;                    ... COMPUTE MAX DWELL TIME FOR SYNCH (24% OF CYCLE) ...
159B 8CC0 A           LD      AC3,AEPD2
159C 8700 A           LD      AC1,D0(AC3)      ; CYCLE LENGTH (SECONDS)
```

```
15A9D 2C48 A          JSR     @AEM24          ; MULTIPLY BY 24
15A9E 4C00 A          LI      AC0,C0          ; PREPARE FOR DIVIDE
15A9F 8ECC A          LD      AC2,AE10        ; *10/100 TO GET 0.1 SEC UNITS
15A0 0690 A           DIV     D0(AC2)
15A1 0000 A
15A2 8CC3 A           LD      AC3,AEPCD2
15A3 A708 A           ST      AC1,D8(AC3)     ; MAX DWELL 24% OF CYCLE (0.1 SEC)
15A4          ;
15A4          ;                  ... COMPUTE START PULSE DURATION (3% OF CYCLE) ...
15A4 50FD A           SHR     AC1,D3          ; DIVIDE BY 8
15A5 A707 A           ST      AC1,D7(AC3)     ; START PULSE DURATION 3% OF CYCLE
15A6 2105 A           JMP     .+6
15A7          ;
15A7          ;                  ... RECORD CYCLE LENGTH VALIDITY ERROR ...
15A7          EPP20:
15A7 8CD2 A           LD      AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
15A8 2C4A A           JSR     @AERE1          ; AC1 = ERROR DATUM
15A9 4C01 A           LI      AC0,C1
15AA 80CE A           ST      AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
15AB 80D0 A           ST      AC0,@AEPDVE     ; SET ERROR FLAG FOR DATA VALIDITY
15AC          ;
15AC          ;                  ... PROCESS OFFSET 1 ...
15AC 88BD A           LD      AC2,AEPD2
15AD 8201 A           LD      AC0,D1(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
15AE 2C46 A           JSR     @AEPBI3         ; CONVERT DATUM TO INTEGER IN AC1
15AF 2110 A           JMP     EPP21           ; DATA VALIDITY ERROR
15B0 8CC0 A           LD      AC3,AEPD2
15B1 A701 A           ST      AC1,D1(AC3)     ; SAVE CONVERTED VALUE (SECONDS)
15B2 E700 A           SKG     AC1,D0(AC3)     ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
15B3 2106 A           JMP     .+7
15B4          ;
15B4          ;                  ... RECORD OFFSET INPUT ERROR ...
15B4 8CD2 A           LD      AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
15B5 2C52 A           JSR     @AERE9          ; AC1 = ERROR DATUM
15B6 4C01 A           LI      AC0,C1
15B7 80CE A           ST      AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
15B8 4DF6 A           LI      AC1,CM10        ; SET OUT OF RANGE RESET TO -1 SECOND
15B9 2103 A           JMP     .+4
15BA          ;
15BA          ;                  ... COMPUTE RESET ...
15BA D700 A           SUB     AC1,D0(AC3)     ; OFFSET - CYCLE
15BB 5101 A           CAI     AC1,C1          ; NEGATE
15BC 2C47 A           JSR     @AEM10          ; MULTIPLY AC1 BY 10
15BD 8CC3 A           LD      AC3,AEPCD2
15BE A704 A           ST      AC1,D4(AC3)     ; SAVE RESET (0.1 SECONDS)
15BF 2105 A           JMP     .+6
15C0          ;
15C0          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
15C0 8CD2 A EPP21:    LD      AC3,AEPEF2      ; BASE ADDRESS OF ERROR BUFFER
15C1 2C4B A           JSR     @AERE2          ; AC1 = ERROR DATUM
15C2 4C01 A           LI      AC0,C1
15C3 80CE A           ST      AC0,@AEP2E      ; SET ERROR FLAG FOR PLAN 2
```

```
15C4 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
15C5          ;
15C5          ;                  ... PROCESS OFFSET 2 ...
15C5 88BD A          LD      AC2,ABPD2
15C6 8202 A          LD      AC0,D2(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
15C7 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
15C8 2110 A          JMP     EPP22            ; DATA VALIDITY ERROR
15C9 8CC0 A          LD      AC3,AEPD2
15CA A702 A          ST      AC1,D2(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
15CB E700 A          SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
15CC 2106 A          JMP     .+7
15CD          ;
15CD          ;                  ... RECORD OFFSET INPUT ERROR ...
15CD 8CD2 A          LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15CE 2C53 A          JSR     @AERE10          ; AC1 = ERROR DATUM
15CF 4C01 A          LI      AC0,C1
15D0 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15D1 4DF6 A          LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
15D2 2103 A          JMP     .+4
15D3          ;
15D3          ;                  ... COMPUTE RESET ...
15D3 D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
15D4 5101 A          CAI     AC1,C1           ; NEGATE
15D5 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
15D6 8CC3 A          LD      AC3,AEPCD2
15D7 A705 A          ST      AC1,D5(AC3)      ; SAVE RESET (0.1 SECONDS)
15D8 2105 A          JMP     .+6
15D9          ;
15D9          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
15D9 8CD2 A EPP22:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15DA 2C4C A          JSR     @AERE3           ; AC1 = ERROR DATUM
15DB 4C01 A          LI      AC0,C1
15DC B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15DD B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
15DE          ;
15DE          ;                  ... PROCESS OFFSET 3 ...
15DE 88BD A          LD      AC2,ABPD2
15DF 8203 A          LD      AC0,D3(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
15E0 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
15E1 2110 A          JMP     EPP23            ; DATA VALIDITY ERROR
15E2 8CC0 A          LD      AC3,AEPD2
15E3 A703 A          ST      AC1,D3(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
15E4 E700 A          SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
15E5 2106 A          JMP     .+7
15E6          ;
15E6          ;                  ... RECORD OFFSET INPUT ERROR ...
15E6 8CD2 A          LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15E7 2C54 A          JSR     @AERE11          ; AC1 = ERROR DATUM
15E8 4C01 A          LI      AC0,C1
15E9 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15EA 4DF6 A          LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
15EB 2103 A          JMP     .+4
```

```
15EC          ;
15EC          ;                    ... COMPUTE RESET ...
15EC D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
15ED 5101 A          CAI     AC1,C1           ; NEGATE
15EE 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
15EF 8CC3 A          LD      AC3,AEPCD2
15F0 A706 A          ST      AC1,D6(AC3)      ; SAVE RESET (0.1 SECONDS)
15F1 2105 A          JMP     .+6
15F2          ;
15F2          ;                    ... RECORD OFFSET DATA VALIDITY ERROR ...
15F2 8CD2 A EPP23:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
15F3 2C4D A          JSR     @AERE4           ; AC1 = ERROR DATUM
15F4 4C01 A          LI      AC0,C1
15F5 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
15F6 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
15F7          ;
15F7          ;                    ... PROCESS MSG SPLIT % ...
15F7 88BD A          LD      AC2,AEPD2
15F8 8204 A          LD      AC0,D4(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
15F9 2C45 A          JSR     @AEPBI2          ; CONVERT DATUM TO INTEGER IN AC1
15FA 2114 A          JMP     EPP24            ; DATA VALIDITY ERROR
15FB 8CC0 A          LD      AC3,AEPD2
15FC A704 A          ST      AC1,D4(AC3)      ; SAVE CONVERTED VALUE (%)
15FD          ;
15FD 88C3 A          LD      AC2,AEPCD2
15FE 8201 A          LD      AC0,D1(AC2)      ; FETCH ADJUSTABLE TIME
15FF 1001 A          BOC     CCZN,.+2         ; TEST FOR POSITIVE
1600 2102 A          JMP     .+3
1601          ;
1601 4000 A          LI      AC1,C0           ; STORE 0 FOR BOTH ADJ. INTS.
1602 2106 A          JMP     .+7
1603          ;
1603 4C00 A          LI      AC0,C0           ; PREPARE FOR MULTIPLY
1604 0680 A          MPY     D1(AC2)          ; MSG(%) * ADJ. TIME (0.1 SEC)
1605 0001 A
1606 88CB A          LD      AC2,AE100        ; DIVIDE BY 100
1607 0690 A          DIV     D0(AC2)
1608 0000 A
1609          ;
1609 88C3 A          LD      AC2,AEPCD2
160A A600 A          ST      AC1,D0(AC2)      ; STORE ADJ. INTERVAL 1
160B 5101 A          CAI     AC1,C1           ; SUBTRACT ADJ. INT. 1 FROM TOTAL
160C C601 A          ADD     AC1,D1(AC2)
160D A601 A          ST      AC1,D1(AC2)      ; STORE ADJ. INT. 2
160E 2105 A          JMP     .+6
160F          ;
160F          ;                    ... RECORD MSG DATA VALIDITY ERROR ...
160F 8CD2 A EPP24:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1610 2C4E A          JSR     @AERE5           ; AC1 = ERROR DATUM
1611 4C01 A          LI      AC0,C1
1612 B0CE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
1613 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
```

```
1614          ;
1614          ;                    ... PROCESS INT1 ...
1614 88BD A          LD      AC2,ABPD2
1615 8205 A          LD      AC0,D5(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1616 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
1617 210E A          JMP     EPP25            ; DATA VALIDITY ERROR
1618 8CC0 A          LD      AC3,AEPD2
1619 A705 A          ST      AC1,D5(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
161A 88CA A          LD      AC2,AEPIN1
161B E600 A          SKG     AC1,D0(AC2)      ; SKIP IF INT1 .GT. LIMIT
161C 2105 A          JMP     .+6
161D          ;
161D          ;                    ... RECORD INT1 INPUT ERROR ...
161D 8CD2 A          LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
161E 2C55 A          JSR     @AERE12          ; AC1 = ERROR DATUM
161F 4C01 A          LI      AC0,C1
1620 BCCE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
1621 94CA A          LD      AC1,@AEPIN1      ; SET MAXIMUM EXTENSION TO LIMIT (SEC)
1622          ;
1622 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
1623 8CC3 A          LD      AC3,AEPCD2
1624 A702 A          ST      AC1,D2(AC3)      ; STORE MAXIMUM EXTENSION (0.1 SEC)
1625 2105 A          JMP     .+6
1626          ;
1626          ;                    ... RECORD INT1 DATA VALIDITY ERROR ...
1626 8CD2 A EPP25:   LD      AC3,AEPEF2       ; BASE ADDRESS OF ERROR BUFFER
1627 2C4F A          JSR     @AERE6           ; AC1 = ERROR DATUM
1628 4C01 A          LI      AC0,C1
1629 BCCE A          ST      AC0,@AEP2E       ; SET ERROR FLAG FOR PLAN 2
162A B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
162B          ;
162B 0200 A          RTS     D0               ; RETURN FROM EPP2
162C          ;
```

```
2480 162C                .LIST   LPP
2481 162C          ;
2482 162C          ;                ... PROCESS PROG. PANEL DATA FOR PLAN 3 ...
2483 162C          EPP3:
2484 162C          ;
2485 162C          ;                ... PROCESS CYCLE LENGTH ...
2486 162C 88BE A         LD      AC2,ABPD3
2487 162D 8200 A         LD      AC0,D0(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
2488 162E 2C46 A         JSR     @AEPBI3          ; CONVERTED VALUE RETURNED IN AC1
2489 162F 212B A         JMP     EPP30            ; DATA VALIDITY ERROR
2490 1630 8CC1 A         LD      AC3,AEPD3
2491 1631 A700 A         ST      AC1,D0(AC3)      ; SAVE CONVERTED INTEGER VALUE (SECONDS)
2492 1632 88C9 A         LD      AC2,AEPCY        ; MAXIMUM CYCLE LENGTH TEST (SECONDS)
2493 1633 E600 A         SKG     AC1,D0(AC2)      ; SKIP IF .GT. MAXIMUM
2494 1634 2107 A         JMP     .+3
2495 1635          ;
2496 1635          ;                ... RECORD CYCLE LENGTH INPUT ERROR ...
2497 1635 8CD3 A         LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
2498 1636 2C51 A         JSR     @AERE8           ; AC1 = ERROR DATUM
2499 1637 4C01 A         LI      AC0,C1
2500 1638 B0CF A         ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
2501 1639 94C9 A         LD      AC1,@AEPCY       ; SET CYCLE LENGTH TO MAXIMUM (SECONDS)
2502 163A 8CC1 A         LD      AC3,AEPD3
2503 163B A700 A         ST      AC1,D0(AC3)      ; SAVE MAXIMUM CYCLE LENGTH (SECONDS)
2504 163C          ;
2505 163C 2C47 A         JSR     @AEM10           ; CONVERT TO 0.1 SECOND UNITS IN AC1
2506 163D 8CC4 A         LD      AC3,AEPCD3
2507 163E A703 A         ST      AC1,D3(AC3)      ; SAVE CYCLE LENGTH (0.1 SEC)
2508 163F          ;
2509 163F 88C5 A         LD      AC2,AEPFT        ; MINIMUM CYCLE LENGTH TEST (0.1 SEC)
2510 1640 8202 A         LD      AC0,D2(AC2)      ; SUM OF FIXED INTERVALS
2511 1641 5001 A         CAI     AC0,C1           ; NEGATE
2512 1642 3400 A         RADD    AC1,AC0          ; AC0 = ADJ. TIME IF NON-NEG.
2513 1643 1B02 A         BOC     CCZN,.+3         ; BRANCH IF CYCLE .LE. MINIMUM
2514 1644 A301 A         ST      AC0,D1(AC3)      ; SAVE ADJUSTABLE TIME IN ADJ. INT. 2
2515 1645 2109 A         JMP     .+10
2516 1646          ;
2517 1646          ;                ... CYCLE LENGTH INPUT ERROR (TOO SHORT) ...
2518 1646 4C00 A         LI      AC0,C0           ; NO ADJUSTABLE TIME
2519 1647 A301 A         ST      AC0,D1(AC3)      ; SAVE 0 IN ADJUSTABLE INTERVAL 2
2520 1648          ;
2521 1648 8200 A         LD      AC0,D0(AC2)      ; RESTORE MIN. SUM OF FIXED INTERVALS
2522 1649 8CC4 A         LD      AC3,AEPCD3
2523 164A A303 A         ST      AC0,D3(AC3)      ; STORE MIN CYCLE (0.1 SEC)
2524 164B 8CD3 A         LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
2525 164C 2C51 A         JSR     @AERE8           ; AC1 = ERROR DATUM
2526 164D 4C01 A         LI      AC0,C1
2527 164E B0CF A         ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
2528 164F          ;
2529 164F          ;                ... COMPUTE MAX DWELL TIME FOR SYNCH (24% OF CYCLE) ...
2530 164F 8CC1 A         LD      AC3,AEPD3
2531 1650 8700 A         LD      AC1,D0(AC3)      ; CYCLE LENGTH (SECONDS)
```

```
1651 2C48 A          JSR     @AEM24          ; MULTIPLY BY 24
1652 4C00 A          LI      AC0,C0          ; PREPARE FOR DIVIDE
1653 88CC A          LD      AC2,AE10        ; *10/100 TO GET 0.1 SEC UNITS
1654 0690 A          DIV     D0(AC2)
1655 0000 A
1656 8CC4 A          LD      AC3,AEPCD3
1657 A700 A          ST      AC1,D8(AC3)     ; MAX DWELL 24% OF CYCLE (0.1 SEC)
1658          ;
1658          ;                  ... COMPUTE START PULSE DURATION (3% OF CYCLE) ...
1658 5DFD A          SHR     AC1,D3          ; DIVIDE BY 8
1659 A707 A          ST      AC1,D7(AC3)     ; START PULSE DURATION 3% OF CYCLE
165A 2105 A          JMP     .+6
165B          ;
165B          ;                  ... RECORD CYCLE LENGTH VALIDITY ERROR ...
165B          EPP30:
165D 8CD3 A          LD      AC3,AEPEF3      ; BASE ADDRESS OF ERROR BUFFER
165C 2C4A A          JSR     @AERE1          ; AC1 = ERROR DATUM
165D 4C01 A          LI      AC0,C1
165E B0CF A          ST      AC0,@AEP3E      ; SET ERROR FLAG FOR PLAN 3
165F B0D0 A          ST      AC0,@AEPDVE     ; SET ERROR FLAG FOR DATA VALIDITY
1660          ;
1660          ;                  ... PROCESS OFFSET 1 ...
1660 88BE A          LD      AC2,ABPD3
1661 8201 A          LD      AC0,D1(AC2)     ; RAW DATUM, BINARY-CODED DIGITS
1662 2C46 A          JSR     @AEPBI3         ; CONVERT DATUM TO INTEGER IN AC1
1663 2110 A          JMP     EPP31           ; DATA VALIDITY ERROR
1664 8CC1 A          LD      AC3,AEPD3
1665 A701 A          ST      AC1,D1(AC3)     ; SAVE CONVERTED VALUE (SECONDS)
1666 E700 A          SKG     AC1,D0(AC3)     ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
1667 2106 A          JMP     .+7
1668          ;
1668          ;                  ... RECORD OFFSET INPUT ERROR ...
1668 8CD3 A          LD      AC3,AEPEF3      ; BASE ADDRESS OF ERROR BUFFER
1669 2C52 A          JSR     @AERE9          ; AC1 = ERROR DATUM
166A 4C01 A          LI      AC0,C1
166B B0CF A          ST      AC0,@AEP3E      ; SET ERROR FLAG FOR PLAN 3
166C 4DF6 A          LI      AC1,CM10        ; SET OUT OF RANGE RESET TO -1 SECOND
166D 2103 A          JMP     .+4
166E          ;
166F          ;                  ... COMPUTE RESET ...
166E D700 A          SUB     AC1,D0(AC3)     ; OFFSET - CYCLE
166F 5101 A          CAI     AC1,C1          ; NEGATE
1670 2C47 A          JSR     @AEM10          ; MULTIPLY AC1 BY 10
1671 8CC4 A          LD      AC3,AEPCD3
1672 A704 A          ST      AC1,D4(AC3)     ; SAVE RESET (0.1 SECONDS)
1673 2105 A          JMP     .+6
1674          ;
1674          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
1674 8CD3 A EPP31:   LD      AC3,AEPEF3      ; BASE ADDRESS OF ERROR BUFFER
1675 2C4B A          JSR     @AERE2          ; AC1 = ERROR DATUM
1676 4C01 A          LI      AC0,C1
1677 B0CF A          ST      AC0,@AEP3E      ; SET ERROR FLAG FOR PLAN 3
```

```
1678 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1679          ;
1679          ;                  ... PROCESS OFFSET 2 ...
1679 88BF A          LD      AC2,ABPD3
167A 8202 A          LD      AC0,D2(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
167B 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
167C 2110 A          JMP     EPP32            ; DATA VALIDITY ERROR
167D 8CC1 A          LD      AC3,AEPD3
167E A702 A          ST      AC1,D2(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
167F E700 A          SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
1680 2106 A          JMP     .+7
1681          ;
1681          ;                  ... RECORD OFFSET INPUT ERROR ...
1681 8CD3 A          LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
1682 2C53 A          JSR     @AERE10          ; AC1 = ERROR DATUM
1683 4C01 A          LI      AC0,C1
1684 B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
1685 4DF6 A          LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
1686 2103 A          JMP     .+4
1687          ;
1687          ;                  ... COMPUTE RESET ...
1687 D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
1688 5101 A          CAI     AC1,C1           ; NEGATE
1689 2C47 A          JSR     @AEMI0           ; MULTIPLY AC1 BY 10
168A 8CC4 A          LD      AC3,AEPCD3
168B A705 A          ST      AC1,D5(AC3)      ; SAVE RESET (0.1 SECONDS)
168C 2105 A          JMP     .+6
168D          ;
168D          ;                  ... RECORD OFFSET DATA VALIDITY ERROR ...
168D 8CD3 A EPP32:   LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
168E 2C4C A          JSR     @AERE3           ; AC1 = ERROR DATUM
168F 4C01 A          LI      AC0,C1
1690 B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
1691 B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
1692          ;
1692          ;                  ... PROCESS OFFSET 3 ...
1692 88BE A          LD      AC2,ABPD3
1693 8203 A          LD      AC0,D3(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
1694 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
1695 2110 A          JMP     EPP33            ; DATA VALIDITY ERROR
1696 8CC1 A          LD      AC3,AEPD3
1697 A703 A          ST      AC1,D3(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
1698 E700 A          SKG     AC1,D0(AC3)      ; SKIP IF OFFSET .GT. CYCLE (SECONDS)
1699 2106 A          JMP     .+7
169A          ;
169A          ;                  ... RECORD OFFSET INPUT ERROR ...
169A 8CD3 A          LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
169B 2C54 A          JSR     @AERE11          ; AC1 = ERROR DATUM
169C 4C01 A          LI      AC0,C1
169D B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
169E 4DF6 A          LI      AC1,CM10         ; SET OUT OF RANGE RESET TO -1 SECOND
169F 2103 A          JMP     .+4
```

```
16A0          ;
16A0          ;                    ... COMPUTE RESET ...
16A0 D700 A          SUB     AC1,D0(AC3)      ; OFFSET - CYCLE
16A1 5101 A          CAI     AC1,C1           ; NEGATE
16A2 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
16A3 8CC4 A          LD      AC3,AEPCD3
16A4 A706 A          ST      AC1,D6(AC3)      ; SAVE RESET (0.1 SECONDS)
16A5 2105 A          JMP     .+6
16A6          ;
16A6          ;                    ... RECORD OFFSET DATA VALIDITY ERROR ...
16A6 8CD3 A EPP33:   LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
16A7 2C4D A          JSR     @AERE4           ; AC1 = ERROR DATUM
16A8 4C01 A          LI      AC0,C1
16A9 BCCF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
16AA BCD0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
16AB          ;
16AB          ;                    ... PROCESS MSG SPLIT % ...
16AB 88BE A          LD      AC2,ABPD3
16AC 8204 A          LD      AC0,D4(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
16AD 2C45 A          JSR     @AEPBI2          ; CONVERT DATUM TO INTEGER IN AC1
16AE 2114 A          JMP     EPP34            ; DATA VALIDITY ERROR
16AF 8CC1 A          LD      AC3,AEPD3
16B0 A704 A          ST      AC1,D4(AC3)      ; SAVE CONVERTED VALUE (%)
16B1          ;
16B1 88C4 A          LD      AC2,AEPCD3
16B2 8201 A          LD      AC0,D1(AC2)      ; FETCH ADJUSTABLE TIME
16B3 1B01 A          BOC     CCZN,.+2         ; TEST FOR POSITIVE
16B4 2102 A          JMP     .+3
16B5          ;
16B5 4D00 A          LI      AC1,C0           ; STORE 0 FOR BOTH ADJ. INTS.
16B6 2106 A          JMP     .+7
16B7          ;
16B7 4C00 A          LI      AC0,C0           ; PREPARE FOR MULTIPLY
16B8 0680 A          MPY     D1(AC2)          ; MSG(%) * ADJ. TIME (0.1 SEC)
     16B9 0001 A
16BA 88C3 A          LD      AC2,AE100        ; DIVIDE BY 100
16BB 0690 A          DIV     D0(AC2)
     16BC 0000 A
16BD          ;
16BD 88C4 A          LD      AC2,AEPCD3
16BE A600 A          ST      AC1,D0(AC2)      ; STORE ADJ. INTERVAL 1
16BF 5101 A          CAI     AC1,C1           ; SUBTRACT ADJ. INT. 1 FROM TOTAL
16C0 C601 A          ADD     AC1,D1(AC2)
16C1 A601 A          ST      AC1,D1(AC2)      ; STORE ADJ. INT. 2
16C2 2105 A          JMP     .+6
16C3          ;
16C3          ;                    ... RECORD MSG DATA VALIDITY ERROR ...
16C3 8CD3 A EPP34:   LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
16C4 2C4E A          JSR     @AERE5           ; AC1 = ERROR DATUM
16C5 4C01 A          LI      AC0,C1
16C6 BCCF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
16C7 BCD0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
```

```
16C8          ;
16C8          ;                    ... PROCESS INT1 ...
16C8 88BE A          LD      AC2,AEPD3
16C9 8205 A          LD      AC0,D5(AC2)      ; RAW DATUM, BINARY-CODED DIGITS
16CA 2C46 A          JSR     @AEPBI3          ; CONVERT DATUM TO INTEGER IN AC1
16CB 210E A          JMP     EPP35            ; DATA VALIDITY ERROR
16CC 8CC1 A          LD      AC3,AEPD3
16CD A705 A          ST      AC1,D5(AC3)      ; SAVE CONVERTED VALUE (SECONDS)
16CE 88CA A          LD      AC2,AEPIN1
16CF E600 A          SKG     AC1,D0(AC2)      ; SKIP IF INT1 .GT. LIMIT
16D0 2105 A          JMP     .+6
16D1          ;
16D1          ;                    ... RECORD INT1 INPUT ERROR ...
16D1 8CD3 A          LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
16D2 2C55 A          JSR     @AERE12          ; AC1 = ERROR DATUM
16D3 4C01 A          LI      AC0,C1
16D4 B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
16D5 94CA A          LD      AC1,@AEPIN1      ; SET MAXIMUM EXTENSION TO LIMIT (SEC)
16D6          ;
16D6 2C47 A          JSR     @AEM10           ; MULTIPLY AC1 BY 10
16D7 8CC4 A          LD      AC3,AEPCD3
16D8 A702 A          ST      AC1,D2(AC3)      ; STORE MAXIMUM EXTENSION (0.1 SEC)
16D9 2105 A          JMP     .+6
16DA          ;
16DA          ;                    ... RECORD INT1 DATA VALIDITY ERROR ...
16DA 8CD3 A EPP35:   LD      AC3,AEPEF3       ; BASE ADDRESS OF ERROR BUFFER
16DB 2C4F A          JSR     @AERE6           ; AC1 = ERROR DATUM
16DC 4C01 A          LI      AC0,C1
16DD B0CF A          ST      AC0,@AEP3E       ; SET ERROR FLAG FOR PLAN 3
16DE B0D0 A          ST      AC0,@AEPDVE      ; SET ERROR FLAG FOR DATA VALIDITY
16DF          ;
16DF 0200 A          RTS     D0               ; RETURN FROM EPP3
16E0          ;
```

```
16E0                    .LIST   LPP
16F0            ;
16F0            ;                   ... UPDATE PROG. PANEL DATA ...
16F0            EPUP:
16F0            ;                   ... UPDATE ADJUSTABLE INTERVALS ...
16F0            ;
16F0            ;                               DIAL 1
16F0 9893 A             LD      AC2,@ATAI1          ; FIRST ADJUSTABLE INTERVAL
16F1 C88B A             ADD     AC2,ATPM1
16F2 8CC2 A             LD      AC3,AEPCD1
16F3 8700 A             LD      AC1,D0(AC3)
16F4 A600 A             ST      AC1,(AC2)
16F5            ;
16F5 9894 A             LD      AC2,@ATAI2          ; SECOND ADJUSTABLE INTERVAL
16F6 C88B A             ADD     AC2,ATPM1
16F7 8701 A             LD      AC1,D1(AC3)
16F8 A600 A             ST      AC1,(AC2)
16F9            ;                               DIAL 2
16F9 9893 A             LD      AC2,@ATAI1          ; FIRST ADJUSTABLE INTERVAL
16FA C88C A             ADD     AC2,ATPM2
16FB 8CC3 A             LD      AC3,AEPCD2
16FC 8700 A             LD      AC1,D0(AC3)
16FD A600 A             ST      AC1,(AC2)
16FE            ;
16FE 9894 A             LD      AC2,@ATAI2          ; SECOND ADJUSTABLE INTERVAL
16FF C88C A             ADD     AC2,ATPM2
1700 8701 A             LD      AC1,D1(AC3)
1701 A600 A             ST      AC1,(AC2)
1702            ;                               DIAL 3
1702 9893 A             LD      AC2,@ATAI1          ; FIRST ADJUSTABLE INTERVAL
1703 C88D A             ADD     AC2,ATPM3
1704 8CC4 A             LD      AC3,AEPCD3
1705 8700 A             LD      AC1,D0(AC3)
1706 A600 A             ST      AC1,(AC2)
1707            ;
1707 9894 A             LD      AC2,@ATAI2          ; SECOND ADJUSTABLE INTERVAL
1708 C88D A             ADD     AC2,ATPM3
1709 8701 A             LD      AC1,D1(AC3)
170A A600 A             ST      AC1,(AC2)
170B            ;
170B            ;                   ... UPDATE MSG SYNCH DWELL PARAMETERS ...
170B 8C89 A             LD      AC3,ATDWL       ; ADDRESS OF MAX DWELL TIMES
170C 88C2 A             LD      AC2,AEPCD1      ; ADDRESS OF CONVERTED DATA FOR DIAL 1
170D 8608 A             LD      AC1,D8(AC2)
170E A700 A             ST      AC1,D0(AC3)
170F            ;
170F 88C3 A             LD      AC2,AEPCD2      ; ADDRESS OF CONVERTED DATA FOR DIAL 2
1710 8608 A             LD      AC1,D8(AC2)
1711 A701 A             ST      AC1,D1(AC3)
1712            ;
1712 88C4 A             LD      AC2,AEPCD3      ; ADDRESS OF CONVERTED DATA FOR DIAL 3
1713 8608 A             LD      AC1,D8(AC2)
```

```
2772 1704 A702 A          ST      AC1,D2(AC3)
2773 1705          ;
2774 1705          ;                  ... UPDATE MAX EXTENSION PREEMPT PARAMETERS ...
2775 1705 88C2 A          LD      AC2,AEPCD1              ; DIAL 1
2776 1706 8602 A          LD      AC1,D2(AC2)
2777 1707 8C76 A          LD      AC3,APEMAX
2778 1708 A701 A          ST      AC1,D1(AC3)
2779 1709          ;
2780 1709 88C3 A          LD      AC2,AEPCD2              ; DIAL 2
2781 170A 8602 A          LD      AC1,D2(AC2)
2782 170B A702 A          ST      AC1,D2(AC3)
2783 170C          ;
2784 170C 88C4 A          LD      AC2,AEPCD3              ; DIAL 3
2785 170D 8602 A          LD      AC1,D2(AC2)
2786 170E A703 A          ST      AC1,D3(AC3)
2787 170F          ;
2788 170F          ;                  ... UPDATE SYNCHRONIZATION PARAMETERS ...
2789 170F 88A1 A          LD      AC2,AESYD
2790 1710 8CC2 A          LD      AC3,AEPCD1              ; DIAL 1
2791 1711 2908 A          JSR     EPUPS
2792 1712          ;
2793 1712 4D06 A          LI      AC1,C6          ; BUFFER LENGTH = 6
2794 1713 3600 A          RADD    AC1,AC2         ; INCREMENT ADDRESS BY BUFFER LENGTH
2795 1714 8CC3 A          LD      AC3,AEPCD2              ; DIAL 2
2796 1715 2904 A          JSR     EPUPS
2797 1716          ;
2798 1716 3600 A          RADD    AC1,AC2         ; INCREMENT ADDRESS BY BUFFER LENGTH
2799 1717 8CC4 A          LD      AC3,AEPCD3              ; DIAL 3
2800 1718 2901 A          JSR     EPUPS
2801 1719          ;
2802 1719 0200 A          RTS     D0              ; RETURN FROM EPUP
2803 171A          ;
2804 171A          ;                                  COPY SYNC DATA ROUTINE
2805 171A 8303 A EPUPS:   LD      AC0,D3(AC3)
2806 171B A200 A          ST      AC0,D0(AC2)     ; CYCLE LENGTH (0.1 SEC UNITS)
2807 171C A205 A          ST      AC0,D5(AC2)     ; SET SYNCH DIAL TIMER TO CYCLE LENGTH
2808 171D          ;
2809 171D 8304 A          LD      AC0,D4(AC3)
2810 171E A201 A          ST      AC0,D1(AC2)     ; RESET 1 (0.1 SEC UNITS)
2811 171F          ;
2812 171F 8305 A          LD      AC0,D5(AC3)
2813 1720 A202 A          ST      AC0,D2(AC2)     ; RESET 2 (0.1 SEC UNITS)
2814 1721          ;
2815 1721 8306 A          LD      AC0,D6(AC3)
2816 1722 A203 A          ST      AC0,D3(AC2)     ; RESET 3 (0.1 SEC UNITS)
2817 1723          ;
2818 1723 8307 A          LD      AC0,D7(AC3)
2819 1724 A204 A          ST      AC0,D4(AC2)     ; START PULSE DURATION (0.1 SEC UNITS)
2820 1725          ;
2821 1725 0200 A          RTS     D0              ; RETURN FROM EPUPS
2822 1726          ;
```

```
1726                    .LIST   LPP
1726          ;                    ... CONVERT BINARY CODED DATA TO INTEGER ...
1726          ;                AT ENTRY, AC0 CONTAINS BINARY CODED DIGITS
1726          ;                ENTRY POINT:
1726          ;                            EPBI3 - AC0 HAS 0,N2,N1,N0
1726          ;                            EPBI2 - AC0 HAS 0, 0,N1,N0
1726          ;                            EPBI1 - AC0 HAS 0, 0, 0,N0
1726          ;                INTEGER VALUE RETURNED IN AC1
1726          EPBI:
1726          ;
1726 4E00 A EPBI3:  LI      AC2,C0
1727 3181 A         RCPY    AC0,AC1                 ; PROCESS THIRD DIGIT (N2)
1728 651E A         AND     AC1,H0F00               ; EXTRACT THIRD DIGIT N2*256
1729 E520 A         SKG     AC1,H0900           ; TEST FOR VALID DIGIT
172A 2101 A         JMP     .+2
172B 211A A         JMP     EPBIR+1             ; ERROR EXIT
172C 5DFE A         SHR     AC1,D2
172D 3600 A         RADD    AC1,AC2                 ; AC2 = AC2 + N2*64
172E 5DFF A         SHR     AC1,D1
172F 3600 A         RADD    AC1,AC2                 ; AC2 = AC2 + N2*(64 + 32)
1730 5DFD A         SHR     AC1,D3
1731 3600 A         RADD    AC1,AC2                 ; AC2 = AC2 + N2*100
1732 2101 A         JMP     .+2
1733 4E00 A EPBI2:  LI      AC2,C0
1734 3181 A         RCPY    AC0,AC1                 ; PROCESS SECOND DIGIT (N1)
1735 6512 A         AND     AC1,H00F0               ; EXTRACT SECOND DIGIT N1*16
1736 E514 A         SKG     AC1,H0090           ; TEST FOR VALID DIGIT
1737 2101 A         JMP     .+2
1738 210D A         JMP     EPBIR+1             ; ERROR EXIT
1739 5DFF A         SHR     AC1,D1
173A 3600 A         RADD    AC1,AC2                 ; AC2 = AC2 + N1*8
173B 5DFE A         SHR     AC1,D2
173C 3600 A         RADD    AC1,AC2                 ; AC2 = AC2 + N1*10
173D 2101 A         JMP     .+2
173E 4E00 A EPBI1:  LI      AC2,C0
173F 3181 A         RCPY    AC0,AC1                 ; PROCESS FIRST DIGIT (N0)
1740 6508 A         AND     AC1,H000F               ; EXTRACT FIRST DIGIT N0*1
1741 E50A A         SKG     AC1,H0009           ; TEST FOR VALID DIGIT
1742 2101 A         JMP     .+2
1743 2102 A         JMP     EPBIR+1             ; ERROR EXIT
1744 3900 A         RADD    AC2,AC1                 ; AC1 = AC2 + N0, FINAL INTEGER
1745          ;
1745 0201 A EPBIR:  RTS     D1                  ; RETURN FROM EPBI
1746 0200 A         RTS     D0                  ; ERROR RETURN FROM EPBI
1747          ;
1747          ;                    ... LOCAL DATA ...
1747 0F00 A H0F00:  .WORD   X'0F00
1748 00F0 A H00F0:  .WORD   X'00F0
1749 000F A H000F:  .WORD   X'000F
174A 0900 A H0900:  .WORD   X'0900              ; MAX VALUE FOR N2*256
174B 0090 A H0090:  .WORD   X'0090              ; MAX VALUE FOR N1*16
174C 0009 A H0009:  .WORD   X'0009              ; MAX VALUE FOR N0*1
```

2877 174D ;

```
174D                    .LIST   LPP
174D            ;
174D            ;
174D            ;
174D            ;                   ... ROUTINES TO MULTIPLY AN INTEGER BY A CONSTANT ...
174D            ;
174D            ;                       UPON ENTRY, AC1 CONTAINS THE INTEGER
174D            ;                                   AC0 USED FOR TEMPORARY STORAGE
174D            ;
174D            ;                       UPON RETURN, AC1 CONTAINS RESULT
174D            ;
174D            ;                   ... MULTIPLY BY 10 ...
174D            EM10:
174D 5D01 A             SHL     AC1,D1          ; INTEGER*2
174E 3481 A             RCPY    AC1,AC0         ; SAVE
174F 5D02 A             SHL     AC1,D2          ; INTEGER*8
1750 3100 A             RADD    AC0,AC1         ; INTEGER*10
1751 0200 A             RTS     D0              ; RETURN FROM EM10
1752            ;
1752            ;                   ... MULTIPLY BY 24 ...
1752            EM24:
1752 5D03 A             SHL     AC1,D3          ; INTEGER*8
1753 3481 A             RCPY    AC1,AC0         ; SAVE
1754 5D01 A             SHL     AC1,D1          ; INTEGER*16
1755 3100 A             RADD    AC0,AC1         ; INTEGER*24
1756 0200 A             RTS     D0              ; RETURN FROM EM24
```

```
 2906 1757                .PAGE
 2907 1757                .ASECT
 2908 1757 1800 A         .=X'1800
 2909 1800        ;
```

```
1800                .LIST   LTP
1800        ;*****************************************************************************
1800        ;
1800        TPEX:
1800 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1801 9484 A         LD      AC1,@ATIN               ; AC1=CURRENT INTERVAL INDEX
1802 8892 A         LD      AC2,ATHI                ;  AC2=LOC OF THI
1803 F600 A         SKNE    AC1,(AC2)               ;SKIP IF NOT TRANSFER/HOLD INTERVAL
1804 2101 A         JMP     .+2
1805 2108 A         JMP     TMSG
1806 9069 A         LD      AC0,@ABDIN              ; AC0 = BDIN
1807 075E A         SKBIT   D14                     ; SKIP IF TRANSFER HOLD
1808 2105 A         JMP     TMSG
1809 8C97 A         LD      AC3,ATCIW               ; AC3=LOC OF COMMAND INDEX WORD TABLE
180A CE00 A         ADD     AC3,(AC2)               ; AC3=LOC OF CIW FOR T/H INTERVAL
180B 8300 A         LD      AC0,(AC3)               ; AC0=CIW FOR T/H INTERVAL
180C B082 A         ST      AC0,@ATCIWC             ; TCIWC=AC0
180D 0200 A         RTS     D0                  ; RETURN FROM TPEX (WAIT MODE)
180E F51E A TMSG:   SKNE    AC1,T0                  ; SKIP IF NOT MSG
180F 2908 A         JSR     TSST                    ; JUMP TO TP SYNC START ROUTINE
1810 2101 A         JMP     .+2                     ; JUMP OVER RTS FOR TIMING MODE
1811 0200 A         RTS     D0                  ; RETURN FROM TPEX (SYNCH DWELL MODE)
1812 9C85 A         LD      AC3,@ATINT              ;  AC2=LOC OF INTERVAL TIMER
1813 DD1A A         SUB     AC3,T1                  ;  DECREMENT INTERVAL TIMER
1814 BC85 A         ST      AC3,@ATINT
1815 ED17 A         SKG     AC3,T0                  ;  SKIP IF INTERVAL NOT DONE
1816 291A A         JSR     TNXT                    ; INITIATE NEXT TP INTERVAL
1817 0200 A         RTS     D0                  ; RETURN FROM TPEX (TIMING MODE)
1818        ;
```

```
1818                .LIST   LTP
1818        ;*****************************************************************************
1818        ;
1818        TSST:
1818 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1819 9C86 A         LD      AC3,@ATFST              ;  AC3=VALUE OF WAIT FOR SYNC START FLAG
181A EC12 A         SKG     AC3,T0                  ; SKIP IF WAIT FOR SYNC START FLAG IS SET
181B 210D A         JMP     TRSET
181C 8889 A         LD      AC2,ATDWL               ; BASE ADDRESS FOR DWELL TIME MAXIMUMS
181D 9483 A         LD      AC1,@ATCDS              ; CURRENT DIAL SELECTION
181E 49FF A         AISZ    AC1,CM1                 ; COMPUTE ADDRESS FOR CURRENT DIAL
181F 3600 A         RADD    AC1,AC2                 ; SKIP IF DIAL 1
1820 8E00 A         LD      AC3,(AC2)               ; MAX DWELL TIME
1821        ;
1821        ;
1821 889C A         LD      AC2,AETST               ; AC2=LOC OF DWELL TIMER
1822 EE00 A         SKG     AC3,(AC2)               ; SKIP IF DWELL TIMER < DWELL MAX
1823 2105 A         JMP     TRSET                   ; JUMP TO RESET WAIT FOR SYNC.START FLAG
1824 4001 A         LI      AC1,C1
1825 889D A         LD      AC2,AEPST               ; AC2=LOC OF START PULSE INDICATOR
1826 4C00 A         LI      AC0,C0
1827 F200 A         SKNE    AC0,(AC2)               ; SKIP IF START PULSE PRESENT
1828 0201 A         RTS     D1                  ; RETURN FROM TSST (SYNCH DWELL MODE)
1829 9C86 A TRSET:  LD      AC0,@ATFST
182A 072F A         SETBIT  D15                     ; SET MSG RELEASE BIT
182B 8C86 A         ST      AC0,@ATFST
182C 0200 A         RTS     D0                  ; RETURN FROM TSST (TIMING MODE)
182D                                              ; LOCAL CONSTANTS
182D 0000 A T0:     .WORD   0
182E 0001 A T1:     .WORD   1
182F 0002 A T2:     .WORD   2
1830 0003 A T3:     .WORD   3
1831        ;
```

```
1831                .LIST   LTP
1831        ;*****************************************************************************
1831        ;
1831        TNXT:
1831 2100 A         JMP     .+1                     ; JSR @ACOTF CAN BE INSERTED HERE
1832 4001 A         LI      AC1,C1
1833 8884 A         LD      AC2,ATIN                ; AC2= LOC OF CURRENT INTERVAL INDEX
1834 C600 A         ADD     AC1,(AC2)
1835 B484 A         ST      AC1,@ATIN               ; INCREMENT INTERVAL INDEX(TIN)
1836 4C00 A         LI      AC0,C0
1837 E200 A         SKG     AC0,(AC2)               ; SKIP IF TIN<0
1838 2101 A         JMP     .+2
1839 211F A         JMP     TOL                     ; TIN < 0
183A 8C95 A         LD      AC3,ATIMX               ; AC3=LOC OF MAX VALUE FOR TIN
183B E700 A         SKG     AC1,(AC3)               ; SKIP IF TIN>MAX
183C 2101 A         JMP     .+2
183D 211B A         JMP     TOL                     ; TIN > TIMX
183E 8C8F A         LD      AC3,ATCTP               ; AC3=LOC OF CURRENT TIMING PLAN
183F CF00 A         ADD     AC3,(AC2)               ; AC3=LOC OF CURRENT INTERVAL IN TP
1840 8300 A         LD      AC0,(AC3)               ; AC0=CURRENT INTERVAL LENGTH
1841 1205 A         BOC     CCZP,TGEZ               ; BRANCH IF INTERVAL => ZERO
1842 4D00 A TLZ:    LI      AC1,C0
1843 B484 A         ST      AC1,@ATIN               ; SET INTERVAL INDEX TO 0
1844 2C2C A         JSR     @AFEOC                  ; JUMP TO END OF CYCLE ROUTINE
1845 8C8F A         LD      AC3,ATCTP               ; AC3=LOC OF CURRENT TP
1846 8300 A         LD      AC0,(AC3)               ; AC0=LENGTH OF FIRST INTERVAL
1847 150A A TGEZ:   BOC     CCNZ,TGZ                ; BRANCH IF AC0 NON-ZERO
1848 2100 A         JMP     .+1                     ; NO OPERATION
1849 2100 A         JMP     .+1                     ; NO OPERATION
184A 2100 A         JMP     .+1                     ; NO OPERATION
184B 8C92 A TEZ:    LD      AC3,ATHI                ; AC3=LOC OF TRANSFER HOLD INDEX
184C F700 A         SKNE    AC1,(AC3)               ; SKIP IF NOT T H INTERVAL
184D 2101 A         JMP     .+2
184E 21E2 A         JMP     TNXT                ; CURRENT INTERVAL  IS 0 AND NOT TRANSFER
184F                                            ; HOLD--JUMP TO NEXT INTERVAL
184F 9069 A         LD      AC0,@ABDIN              ; AC0=BDIN
1850 075E A         SKBIT   D14                     ; SKIP IF TRANSFER HOLD
1851 21DF A         JMP     TNXT                    ; NO TRANSFER HOLD, SKIP TO NEXT INTERVAL
1852 B085 A TGZ:    ST      AC0,@ATINT              ; INITIALIZE CURRENT INTERVAL TIMER
1853 8897 A         LD      AC2,ATCIW               ; AC2=LOC OF COMMAND INDEX WORD TABLE
1854 9C84 A         LD      AC3,@ATIN               ; AC3=VALUE OF CURRENT INT INDEX
1855 3E00 A         RADD    AC3,AC2                 ; AC2=LOC OF CURRENT COMMAND INDEX WORD
1856 8200 A         LD      AC0,(AC2)               ; AC0=VALUE OF COMMAND INDEX WORD
1857 B082 A         ST      AC0,@ATCIWC             ;TCIWC=VALUE OF COMMAND INDEX WORD
1858 0200 A         RTS     D0                  ; RETURN FROM TNXT
1859        TOL:                                ; INTERVAL INDEX OUT OF LIMITS, RECORD ERROR
1859                                            ;     AND DISCONNECT
1859 8890 A         LD      AC2,ATERR
185A 9084 A         LD      AC0,@ATIN
185B A204 A         ST      AC0,D4(AC2)             ; STORE ERROR WORD IN ERROR BUFFER
185C 8203 A         LD      AC0,D3(AC2)
185D 4801 A         AISZ    AC0,C1                  ; FETCH ERROR COUNTER AND INCREMENT
```

TIMING PLAN CPC - INITIATE NEXT INTERVAL

```
3030 185E 2100 A        JMP     .+1
3031 185F A203 A        ST      AC0,D3(AC2)
3032 1860 8200 A        LD      AC0,D0(AC2)
3033 1861 0721 A        SETBIT  D1                      ; SET ERROR FLAG BIT 1
3034 1862 A200 A        ST      AC0,D0(AC2)
3035 1863       ;
3036 1863 2C2F A        JSR     @AEADC          ; AUTODISCONNECT ROUTINE, NO RETURN
3037 1864       ;
3038 1864 2100 A        JMP     .+1             ; THIS INST SHOULD NEVER BE REACHED
3039 1865 21FE A        JMP     .-1
3040 1866       ;
```

```
1866                    .LIST   LTP
1866          ;*****************************************************************************
1866          ;
1866          TNCY:
1866          ;                                         ZERO PRED AND PTER ENTRY TIMES
1866 4C00 A           LI      AC0,C0
1867 889A A           LD      AC2,ACTST
1868 A200 A           ST      AC0,D0(AC2)
1869 A201 A           ST      AC0,D1(AC2)
186A          ;
186A          ;                                         TEST PREEMPT IN PROGRESS FLAG
186A 906E A           LD      AC0,@APIP                 ; ACCESS PREEMPT IN PROGRESS FLAG
186B 1554 A           BOC     CCNZ,TNDC                 ; BRANCH IF PREEMPT IN PROGRESS
186C                                                    ;    NO DIAL CHANGE PERMITTED
186C          ;
186C          ;                    ... MANUAL OVERRIDE TEST ...
186C 90BB A           LD      AC0,@ABPPD        ; PROG PANEL SWITCHES
186D 075B A           SKBIT   D11               ; SKIP IF MANUAL OVERRIDE IS ON
186E 2101 A           JMP     .+2               ; MANUAL OVERRIDE OFF
186F 2116 A           JMP     TSYNCR
1870          ;
1870          ;                    ... PROG PANEL SYNCHRONIZATION SWITCHES ...
1870 8883 A TSYNC:    LD      AC2,ATCDS         ; ADDRESS OF CURRENT DIAL SELECT DATUM
1871          ;
1871 4D01 A           LI      AC1,C1
1872 F600 A           SKNE    AC1,(AC2)         ; SKIP IF NOT DIAL 1
1873 2101 A           JMP     .+2
1874 2103 A           JMP     TSD2              ; JUMP TO TEST DIAL 2
1875 0751 A           SKBIT   D1                ; SKIP IF SYNCH SWITCH IS ON
1876 210F A           JMP     TSYNCR            ; JUMP TO RESET FLAG
1877 2110 A           JMP     TSYNCE
1878          ;
1878 4D02 A TSD2:     LI      AC1,C2
1879 F600 A           SKNE    AC1,(AC2)         ; SKIP IF NOT DIAL 2
187A 2101 A           JMP     .+2
187B 2103 A           JMP     TSD3              ; JUMP TO TEST DIAL 3
187C 0755 A           SKBIT   D5                ; SKIP IF SYNCH SWITCH IS ON
187D 2108 A           JMP     TSYNCR            ; JUMP TO RESET FLAG
187E 2109 A           JMP     TSYNCE
187F          ;
187F 4D03 A TSD3:     LI      AC1,C3
1880 F600 A           SKNE    AC1,(AC2)         ; SKIP IF NOT DIAL 3
1881 2101 A           JMP     .+2
1882 2105 A           JMP     TSYNCE
1883 0759 A           SKBIT   D9                ; SKIP IF SYNCH SWITCH IS ON
1884 2101 A           JMP     TSYNCR            ; JUMP TO RESET FLAG
1885 2102 A           JMP     TSYNCE
1886          ;
1886 4C00 A TSYNCR:   LI      AC0,C0
1887 B086 A           ST      AC0,@ATFST        ; RESET 'WAIT FOR SYNCH START' FLAG
1888          ;
1888          TSYNCE:
```

```
1888 906E A          LD      AC0,@APIP        ; FETCH PREEMPT IN PROGRESS FLAG
1889 150C A          BOC     CCNZ,TPUE        ; BRANCH IF FLAG SET
188A 90C6 A          LD      AC0,@AEPUF       ; FETCH PROG PANEL UPDATE FLAG
188B 1501 A          BOC     CCNZ,.+2         ; BRANCH IF UPDATE FLAG IS SET
188C 2109 A          JMP     TPUE
188D          ;
188D 2C44 A          JSR     @AEPUP           ; UPDATE PROG. PANEL DATA
188E          ;
188E 8CE9 A          LD      AC3,APPDU
188F 2C57 A          JSR     @AXLOF1          ; ASSURE PENDING INPUT LIGHT IS OFF
1890 4C00 A          LI      AC0,C0
1891 B0C6 A          ST      AC0,@AEPUF       ; RESET PROG PANEL UPDATE FLAG
1892          ;
1892 4C00 A          LI      AC0,C0
1893 B086 A          ST      AC0,@ATFST       ; RESET 'WAIT FOR SYNCH START' FLAG
1894          ;
1894 9483 A          LD      AC1,@ATCDS       ; TREAT PANEL UPDATE AS A DIAL CHANGE
1895 211E A          JMP     TDCH
1896          TPUE:
1896          ;
1896                                          ; DETERMINE CURRENT DIAL SELECTION
1896 90A4 A TDIAL:   LD      AC0,@AED23
1897 0754 A          SKBIT   D4
1898 2113 A          JMP     TND2                     ; NOT DIAL 2
1899 0755 A          SKBIT   D5
189A 210F A          JMP     TD2                      ; SELECT DIAL 2
189B          ;                                   DIAL SELECT ERROR
189B 8890 A          LD      AC2,ATERR                ; IF BOTH BITS 4 AND 5 ARE SET, RECORD
189C                                                  ; ERROR
189C A202 A          ST      AC0,D2(AC2)              ; STORE IN ERROR BUFFER TERRO
189D 8201 A          LD      AC0,D1(AC2)              ; FETCH ERROR COUNTER AND INCREMENT
189E 4801 A          AISZ    AC0,C1
189F 2100 A          JMP     .+1
18A0 A201 A          ST      AC0,D1(AC2)
18A1 8200 A          LD      AC0,D0(AC2)
18A2 0720 A          SETBIT  D0                       ; SET ERROR FLAG BIT 0
18A3 A200 A          ST      AC0,D0(AC2)
18A4 9483 A          LD      AC1,@ATCDS               ; IN CASE OF ERROR, RETAIN CURRENT DIAL
18A5 E588 A          SKG     AC1,T1                   ; SKIP IF NOT DIAL 1
18A6 2109 A          JMP     TD1
18A7 E587 A          SKG     AC1,T2                   ; SKIP IF DIAL 3
18A8 2101 A          JMP     TD2
18A9 2104 A          JMP     TD3
18AA 4D02 A TD2:     LI      AC1,C2                   ; SELECT DIAL 2
18AB 2105 A          JMP     TDSEL
18AC 0755 A TND2:    SKBIT   D5                       ; SKIP IF DIAL 3
18AD 2102 A          JMP     TD1
18AE 4D03 A TD3:     LI      AC1,C3                   ; SELECT DIAL 3
18AF 2101 A          JMP     TDSEL
18B0 4D01 A TD1:     LI      AC1,C1                   ; SELECT DIAL 1
18B1          ;
18B1          TDSEL:
```

TIMING PLAN CPC - INITIATE NEXT CYCLE

```
3147 18B1 8C83 A          LD      AC3,ATCDS          ; AC3=LOC OF CURRENT DIAL SELECTOR
3148 18B2 F700 A          SKNE    AC1,(AC3)          ; SKIP IF NEW DIAL SELECTION
3149 18B3 210C A          JMP     TNDC               ; JUMP NO DIAL CHANGE
3150 18B4         ;                                DIAL CHANGE
3151 18B4 B483 A TDCH:    ST      AC1,@ATCDS         ; CHANGE CURRENT DIAL SELECTOR
3152 18B5 9C83 A          LD      AC3,@ATCDS         ; UPDATED DIAL SELECTION (1, 2, OR 3)
3153 18B6 CC8A A          ADD     AC3,ATPP           ; ADDRESS PRECEDING POINTERS
3154 18B7 8B00 A          LD      AC2,D0(AC3)        ; POINTER TO MASTER FOR CURRENT DIAL
3155 18B8 8C8F A          LD      AC3,ATCTP          ; AC3=LOC OF CURRENT TP TABLE
3156 18B9 2C36 A          JSR     @ATCOPY            ; COPY MASTER INTO CURRENT
3157 18BA 9C83 A          LD      AC3,@ATCDS         ; UPDATED DIAL SELECTION (1, 2, OR 3)
3158 18BB CC8A A          ADD     AC3,ATPP           ; ADDRESS PRECEDING POINTERS
3159 18BC 8B00 A          LD      AC2,D0(AC3)        ; POINTER TO MASTER FOR CURRENT DIAL
3160 18BD 8C8E A          LD      AC3,ATNTP          ; AC3=LOC OF NEXT TP
3161 18BE 2C36 A          JSR     @ATCOPY            ;COPY MASTER TO NEXT
3162 18BF 0200 A          RTS     D0                 ; RETURN FROM TNCY
3163 18C0         ;
3164 18C0         ;                                NO DIAL CHANGE
3165 18C0 8C8F A TNDC:    LD      AC3,ATCTP
3166 18C1 888E A          LD      AC2,ATNTP
3167 18C2 2C36 A          JSR     @ATCOPY            ; COPY NEXT TP INTO CURRENT TP
3168 18C3 9C83 A          LD      AC3,@ATCDS         ; UPDATED DIAL SELECTION (1, 2, OR 3)
3169 18C4 CC8A A          ADD     AC3,ATPP           ; ADDRESS PRECEDING POINTERS
3170 18C5 8B00 A          LD      AC2,D0(AC3)        ; POINTER TO MASTER FOR CURRENT DIAL
3171 18C6 8C8E A          LD      AC3,ATNTP
3172 18C7 2C36 A          JSR     @ATCOPY            ;COPY MASTER TO NEXT
3173 18C8 0200 A          RTS     D0               ; RETURN FROM TNCY
3174 18C9         ;
```

```
3177 18C9                .LIST   LTP
3178 18C9        ;*****************************************************************************
3179 18C9        ;
3180 18C9        ;                                TCOPY COPIES ONE 20 ELEMENT ARRAY INTO
3181 18C9        ;                                ANOTHER 20 ELEMENT ARRAY
3182 18C9        ;                                AC3 PASSES THE LOCATION OF ARRAY TO BE
3183 18C9        ;                                CHANGED.  AC2 PASSES THE LOCATION OF
3184 18C9        ;                                THE ARRAY TO BE COPIED
3185 18C9        ;*****************************************************************************
3186 18C9 4DEC A TCOPY:  LI      AC1,CM20
3187 18CA 8200 A TLOOP:  LD      AC0,(AC2)              ;VALUE OF THE OLD ARRAY
3188 18CB A300 A         ST      AC0,(AC3)              ; LOC AC3 IN NEW TABLE = AC0
3189 18CC 4901 A         AISZ    AC1,C1
3190 18CD 2101 A         JMP     .+2
3191 18CE 0200 A         RTS     D0                 ; RETURN FROM TCOPY
3192 18CF 4A01 A         AISZ    AC2,C1
3193 18D0 4B01 A         AISZ    AC3,C1
3194 18D1 21F8 A         JMP     TLOOP
3195 18D2 0200 A         RTS     D0                 ; RETURN FROM TCOPY
3196 18D3        ;
```

```
18D3                    .LIST   LTP
18D3            ;
18D3            ;                   ... T I M I N G   P L A N   T A B L E S ...
18D3            ;
18D3            ;                       (ALL DATA IN THIS LOCAL DATA BASE SHALL BE
18D3            ;                        CONSIDERED AS APPLICATION DEPENDENT)
18D3            ;
18D3            ;                       PASSIVE BUS DETECTOR SYSTEM - PBD14C
18D3            ;
18D3            ;       (INTERVAL INDEX PARAMETERS ARE IN TERMS OF DISPLACEMENT FROM
18D3            ;        THE BEGINNING OF THE TIMING PLAN INTERVAL TABLE, 0 - N)
18D3            ;
18D3 0009 A TSCI:       .WORD   9               ; INDEX OF SYNCH CONTRACTIBLE INTERVAL
18D4 000A A THI:        .WORD   10              ; INDEX OF TRANSFER/HOLD INTERVAL
18D5 0005 A TAI1:       .WORD   5               ; INDEX OF ADJ INT 1 (MUST BE IN MSG)
18D6 0009 A TAI2:       .WORD   9               ; INDEX OF ADJ INT 2
18D7 000F A TIMX:       .WORD   15              ; MAXIMUM INTERVAL INDEX
18D8            ;
18D8            ;                   ... PARAMETERS FOR DETECTORIZED APPROACH ...
18D8            ;
18D8            TECI:
18D8            ;
18D8            ;
18D8            ;
18D8 0000 A             .WORD   0               ; NOT USED
18D9 000A A             .WORD   10              ; FIRST INTERVAL IN EXTENSION REQUEST RANGE
18DA 0002 A             .WORD   2               ; EXTENDIBLE INTERVAL INDEX
18DB 003C A             .WORD   60              ; INCREMENTAL EXTENSION TIME
18DC 0005 A             .WORD   5               ; PRIMARY CONTRACTIBLE INTERVAL INDEX
18DD 0009 A             .WORD   9               ; SECONDARY CONTRACTIBLE INTERVAL INDEX
18DE FFFF A             .WORD   -1              ; TERTIARY CONTRACTIBLE INTERVAL INDEX
18DF 0002 A             .WORD   TLEP-TFEP+1     ; NUMBER OF INTERVAL PAIRS TO BE EXCHANGED
18E0            TFEP:
18E0 0002 A             .WORD   X'0002          ; INDICES OF EXCHANGE PAIR 1
18E1 0103 A TLEP:       .WORD   X'0103          ; INDICES OF EXCHANGE PAIR 2
18E2 1918 A             .=.+54
1918            ;
1918            ;                   ... SIGNAL DISPLAY COMMAND INDEX WORDS ...
1918            ;                           (MAXIMUM OF 20 INTERVALS)
1918            ;
1918            ;                       (BITS 15 - 8, MAIN STREET COMMAND INDEX
1918            ;                        BITS 7 - 0, CROSS STREET COMMAND INDEX)
1918            TCIW:
1918 060E A             .WORD   X'060E          ; INTERVAL 1
1919 060E A             .WORD   X'060E          ; INTERVAL 2
191A 080E A             .WORD   X'080E          ; INTERVAL 1A
191B 0C0E A             .WORD   X'0C0E          ; INTERVAL 2A
191C 060E A             .WORD   X'060E          ; INTERVAL 3B
191D 060E A             .WORD   X'060E          ; INTERVAL 3
191E 070E A             .WORD   X'070E          ; INTERVAL 4
191F 080E A             .WORD   X'080E          ; INTERVAL 5
1920 090E A             .WORD   X'090E          ; INTERVAL 6B
```

```
1921 090E A        .WORD   X'090E          ; INTERVAL 6
1922 0A0E A        .WORD   X'0A0E          ; INTERVAL 7
1923 0D0F A        .WORD   X'0D0F          ; INTERVAL 8
1924 0D10 A        .WORD   X'0D10          ; INTERVAL 9
1925 0D11 A        .WORD   X'0D11          ; INTERVAL 10
1926 0D0E A        .WORD   X'0D0E          ; INTERVAL 11
1927          ;
1927 192C A        .=.+5
192C          ;
192C          ;                    ... SKELETON TIMING PLAN TABLES ...
192C          ;                          ( ADJ INTS = 0)
192C          ;
192C          TPTAB:
192C 0046 A        .WORD   70              ; TPM1
192D 0028 A        .WORD   40
192E 0000 A        .WORD   0
192F 0000 A        .WORD   0
1930 000A A        .WORD   10
1931 0000 A        .WORD   0               ; ADJ INT 1
1932 0078 A        .WORD   120
1933 0028 A        .WORD   40
1934 0046 A        .WORD   70
1935 0000 A        .WORD   0               ; ADJ INT 2
1936 0028 A        .WORD   40
1937 004B A        .WORD   75
1938 0091 A        .WORD   145
1939 0028 A        .WORD   40
193A 000F A        .WORD   15
193B FFFF A        .WORD   -1
193C 1940 A        .=.+4
1940          ;
1940 0046 A        .WORD   70              ; TPM2
1941 0028 A        .WORD   40
1942 0000 A        .WORD   0
1943 0000 A        .WORD   0
1944 000A A        .WORD   10
1945 0000 A        .WORD   0               ; ADJ INT 1
1946 0078 A        .WORD   120
1947 0028 A        .WORD   40
1948 0046 A        .WORD   70
1949 0000 A        .WORD   0               ; ADJ INT 2
194A 0028 A        .WORD   40
194B 004B A        .WORD   75
194C 0091 A        .WORD   145
194D 0028 A        .WORD   40
194E 000F A        .WORD   15
194F FFFF A        .WORD   -1
1950 1954 A        .=.+4
1954          ;
1954 0046 A        .WORD   70              ; TPM3
1955 0028 A        .WORD   40
1956 0000 A        .WORD   0
```

```
3303 1957 0000 A          .WORD   0
3304 1958 000A A          .WORD   10
3305 1959 0000 A          .WORD   0                ; ADJ INT 1
3306 195A 0078 A          .WORD   120
3307 195B 0028 A          .WORD   40
3308 195C 0046 A          .WORD   70
3309 195D 0000 A          .WORD   0                ; ADJ INT 2
3310 195E 0028 A          .WORD   40
3311 195F 004B A          .WORD   75
3312 1960 0091 A          .WORD   145
3313 1961 0028 A          .WORD   40
3314 1962 000F A          .WORD   15
3315 1963 FFFF A          .WORD   -1
3316 1964 1967 A          .=.+3
3317 1967 1968 A TPE:     .=.+1
3318 1968        ; END OF TP TABLES
```

```
3319 1968                .PAGE
3320 1968                .ASECT
3321 1968 1A00 A         .=X'1A00
```

```
1A00                    .LIST   LPR
1A00            PMON:
1A00 2100 A             JMP     .+1
1A01            ;                                   DECREMENT RECENT BUS CLASSIFICATION TIMER
1A01 907B A             LD      AC0,@AVCBT
1A02 1103 A             BOC     CCEZ,.+4                ; BRANCH IF TIMER = 0
1A03 48FF A             AISZ    AC0,CM1                 ; DECREMENT TIMER
1A04 2100 A             JMP     .+1                     ; NO OPERATION
1A05 B07B A             ST      AC0,@AVCBT
1A06            ;                                   SHOULD PIP BE RESET
1A06 9084 A             LD      AC0,@ATIN               ; AC0 = CURRENT INTERVAL INDEX
1A07 8896 A             LD      AC2,ATECI
1A08 E202 A             SKG     AC0,D2(AC2)             ; SKIP IF CURRENT INTERVAL IS AFTER
1A09            ;                                         EXTENDED INTERVAL
1A09 2105 A             JMP     PM0
1A0A E201 A             SKG     AC0,D1(AC2)             ; SKIP IF CURRENT INTERVAL PRIOR
1A0B            ;                                         TO REQUEST RANGE
1A0B F201 A             SKNE    AC0,D1(AC2)
1A0C 2102 A             JMP     PM0
1A0D            ;                                   RESET PIP FLAG
1A0D 4C00 A             LI      AC0,C0
1A0E B06E A             ST      AC0,@APIP
1A0F            PM0:
1A0F            ;                                   CLASSIFY VEHICLE SIGNATURE
1A0F 4C01 A             LI      AC0,C1
1A10            PM01:
1A10 B07E A             ST      AC0,@AVCDT              ; STORE DETECTOR INDEX
1A11 2C31 A             JSR     @AVCLAS
1A12 1501 A             BOC     CCNZ,.+2                ; BRANCH IF VEHICLE NON-BUS
1A13 2106 A             JMP     PM02                    ; JUMP TO HANDLE BUS
1A14 907E A             LD      AC0,@AVCDT
1A15 4801 A             AISZ    AC0,C1                  ; INCREMENT DETECTOR INDEX WORD
1A16 2100 A             JMP     .+1                     ; NO OPERATION
1A17 E136 A             SKG     AC0,PMHD                ; SKIP IF ALL DETECTOR BLOCKS HAVE
1A18            ;                                         BEEN EXAMINED
1A18 21F7 A             JMP     PM01                    ; EXAMINE NEXT DETECTOR BLOCK
1A19 0200 A             RTS     D0                      ; RETURN FROM PMON
1A1A            PM02:
1A1A            ;                                   TEST PROGRAM PANEL BUS PREEMPT SWITCH
1A1A 9C83 A             LD      AC3,@ATCDS              ; AC3 = DIAL SETTING
1A1B CD2C A             ADD     AC3,PMAS
1A1C 90BB A             LD      AC0,@ABPPD
1A1D 7300 A             SKAZ    AC0,D0(AC3)             ; SKIP IF SWITCH IS OFF
1A1E 2101 A             JMP     .+2
1A1F 0200 A             RTS     D0                      ; RETURN
1A20            ;                                   BUS PREEMPT SWITCH IS ON
1A20            ;                                   IS CURRENT INTERVAL WITHIN EXTENSION
1A20            ;                                         REQUEST RANGE
1A20 9084 A             LD      AC0,@ATIN
1A21 8896 A             LD      AC2,ATECI
1A22 E201 A             SKG     AC0,D1(AC2)             ; SKIP IF INTERNAL WITHIN REQUEST RANGE
1A23 F201 A             SKNE    AC0,D1(AC2)             ; SKIP IF INTERVAL NOT WITHIN
```

```
1A24          ;                                          REQUEST RANGE
1A24 2101 A          JMP     .+2
1A25 212A A          JMP     PM1
1A26          ;                                  IF PREEMPT IN PROGRESS, RETURN.
1A26 906E A          LD      AC0,@APIP                  ; AC0 = PREEMPT IN PROGRESS FLAG
1A27 151F A          BOC     CCNZ,PMRET                 ; BRANCH IF PREEMPT IN PROGRESS
1A28          ;                                  SCHEDULE BUS INTERVAL NEXT CYCLE
1A28 8896 A          LD      AC2,ATECI
1A29 8607 A          LD      AC1,D7(AC2)                ; AC1 = NUMBER OF EXCHANGE PAIRS
1A2A 3600 A          RADD    AC1,AC2                    ; AC2 = BASE ADDRESS FOR EXCHANGE
1A2B          PM05:
1A2B 8E07 A          LD      AC3,D7(AC2)                ; AC3 = EXCHANGE PAIR INDEX WORD
1A2C          ;                                  COMPUTE ADDR POINTER TO NON-BUS INTERVAL
1A2C 5FF8 A          SHR     AC3,D8
1A2D 811E A          LD      AC0,PMAS2
1A2E 3383 A          RAND    AC0,AC3
1A2F CC8E A          ADD     AC3,ATNTP
1A30          ;                                  IF DURATION OF THIS INTERVAL IS <= ZERO,
1A30          ;                                  BRANCH TO ERROR HANDLING ROUTINE
1A30 8300 A          LD      AC0,D0(AC3)
1A31 1812 A          BOC     CCZN,PMER10
1A32          ;                                  SET DURATION OF NON-BUS INTERVAL TO ZERO
1A32 B06F A          ST      AC0,@APTEM
1A33 4C00 A          LI      AC0,C0
1A34 A300 A          ST      AC0,D0(AC3)
1A35          ;                                  COMPUTE ADDR POINTER TO BUS INTERVAL
1A35 8E07 A          LD      AC3,D7(AC2)                ; AC3 = EXCHANGE PAIR INDEX WORD
1A36 8115 A          LD      AC0,PMAS2
1A37 3383 A          RAND    AC0,AC3
1A38 CC8E A          ADD     AC3,ATNTP
1A39          ;                                  TRANSFER NON-BUS DURATION TO BUS INTERVAL
1A39 906F A          LD      AC0,@APTEM
1A3A A300 A          ST      AC0,D0(AC3)
1A3B          ;                                  DECREMENT BASE ADDRESS AND EXCHANGE
1A3B          ;                                  PAIR COUNTER
1A3B 4AFF A          AISZ    AC2,CM1                    ; SHOULD NOT SKIP
1A3C 49FF A          AISZ    AC1,CM1                    ; SKIP IF EXCHANGES COMPLETED
1A3D 21ED A          JMP     PM05                       ; JUMP TO CONTINUE EXCHANGES
1A3E          ;                                  SET FLAG FOR THE EXTENDIBLE PREEMPT
1A3E 4C01 A          LI      AC0,C1
1A3F B06E A          ST      AC0,@APIP
1A40 B074 A          ST      AC0,@APFREQ
1A41          ;                                  RESET WAIT FOR SYNCH START FLAG
1A41 4C00 A          LI      AC0,C0
1A42 B086 A          ST      AC0,@ATFST
1A43 0200 A          RTS     D0                         ; RETURN FROM PMON
1A44 3981 A PMER10:  RCPY    AC2,AC1                    ; SAVE ADDR OF ZEROED INTERVAL
1A45 8C77 A          LD      AC3,APERR
1A46 2C52 A          JSR     @AERE9                     ; JUMP TO ERROR HANDLING ROUTINE
1A47 0200 A PMRET:   RTS     D0                         ; RETURN
```

```
3426 1A48                 .PAGE 'LOCAL DATA'
3427 1A48 1A48 A PMAS:    .WORD   PMAS
3428 1A49 0001 A          .WORD   X'0001                    ; LOCAL DATA
3429 1A4A 0010 A          .WORD   X'0010
3430 1A4B 0100 A          .WORD   X'0100
3431 1A4C 00FF A PMAS2:   .WORD   X'00FF
3432 1A4D FFFF A PMM1:    .WORD   -1
3433 1A4E 0001 A PMHD:    .WORD   DETS
3434 1A4F 1A98 A APSWCH:  .WORD   PSWCH
```

```
1A50                    .PAGE 'PREEMPT MONITOR - CONTINUED'
1A50 F202 A PM1:        SKNE    AC0,D2(AC2)          ; SKIP IF CURRENT INTERVAL INDEX NOT =
1A51                                                 ; EI INDEX
1A51 2101 A             JMP     PM2                  ; JUMP TO EXTEND EI INTERVAL
1A52 0200 A             RTS     D0                   ; RETURN
1A53            PM2:
1A53            ;                                 DETERMINE PEMAX FOR CURRENT DIAL SETTING
1A53 9C83 A             LD      AC3,@ATCDS           ; AC3 = CURRENT DIAL SETTING
1A54 CC76 A             ADD     AC3,APEMAX           ; AC3 = ADDR POINTER TO VALUE OF E(MAX)
1A55            ;                                      FOR CURRENT DIAL SETTING
1A55 8300 A             LD      AC0,D0(AC3)          ; AC0 = E(MAX) FOR CURRENT DIAL SETTING
1A56 B076 A             ST      AC0,@APEMAX
1A57            ;                                 DETERMINE DELTA E FOR THIS EXTENSION ATTEMPT
1A57 8C8F A             LD      AC3,ATCTP
1A58 CE02 A             ADD     AC3,D2(AC2)          ; AC3 POINTS AT EI
1A59 8300 A             LD      AC0,D0(AC3)          ; AC0 = CURRENT DURATION OF EI
1A5A C203 A             ADD     AC0,D3(AC2)          ; AC0 = AC0 + STANDARD DELTA E
1A5B 8876 A             LD      AC2,APEMAX
1A5C E200 A             SKG     AC0,D0(AC2)          ; SKIP IF AC0 > PEMAX
1A5D 2106 A             JMP     .+7
1A5E 8200 A             LD      AC0,D0(AC2)          ; AC0 = PEMAX
1A5F D300 A             SUB     AC0,D0(AC3)          ; AC0 = PEMAX - DURATION OF EI
1A60 8896 A             LD      AC2,ATECI
1A61 1204 A             BOC     CCZP,.+5
1A62 4C00 A             LI      AC0,C0               ; DELTA E = 0
1A63 2102 A             JMP     .+3
1A64 8896 A             LD      AC2,ATECI
1A65 8203 A             LD      AC0,D3(AC2)
1A66 B06F A             ST      AC0,@APTEM        ; SET DELTA E
1A67 8C8F A             LD      AC3,ATCTP
1A68 CE04 A             ADD     AC3,D4(AC2)          ; AC3 POINTS TO CI(P)
1A69 8300 A             LD      AC0,D0(AC3)          ; AC0 = DURATION OF CI(P)
1A6A 886F A             LD      AC2,APTEM
1A6B E200 A             SKG     AC0,D0(AC2)          ; SKIP IF DURATION OF CI(P) > DELTA E
1A6C F200 A             SKNE    AC0,D0(AC2)          ;SKIP IF AC0 NOT = DELTA E
1A6D 211D A             JMP     PM3
1A6E            ;                                 SEEK ADDITIONAL TIME FROM CI(S)
1A6E 4D00 A             LI      AC1,C0               ; SET DURATION OF CI(P) TO ZERO
1A6F A700 A             ST      AC1,D0(AC3)
1A70 8896 A             LD      AC2,ATECI
1A71 8605 A             LD      AC1,D5(AC2)          ; AC1 = INDICE TO SECONDARY CI
1A72 E5DA A             SKG     AC1,PMM1             ; SKIP IF INDICE => ZERO
1A73 211A A             JMP     PM5                  ; NO SECONDARY CI, USE TIME AVAILABLE
1A74 8C8F A             LD      AC3,ATCTP
1A75 CE05 A             ADD     AC3,D5(AC2)          ; AC3 POINTS TO CI(S)
1A76            ;                                 DETERMINE ADDITIONAL TIME NEEDED
1A76 886F A             LD      AC2,APTEM
1A77 D200 A             SUB     AC0,D0(AC2)
1A78 C300 A             ADD     AC0,D0(AC3)
1A79 1212 A             BOC     CCZP,PM4             ; BRANCH IF DELTA E TIME AVAILABLE
1A7A C200 A             ADD     AC0,D0(AC2)
1A7B 4D00 A             LI      AC1,C0               ; SET DURATION OF CI(S) TO ZERO
```

```
1A7C A700 A        ST      AC1,D0(AC3)
1A7D        ;                                  SEEK ADDITIONAL TIME FROM CI(T)
1A7D 8896 A        LD      AC2,ATECI
1A7E 8606 A        LD      AC1,D6(AC2)            ; AC1 = INDICE TO TERTIARY CI
1A7F E5CD A        SKG     AC1,PMM1               ; SKIP IF INDICE => THAN ZERO
1A80 210D A        JMP     PM5                    ; NO TERTIARY CI, USE AVAILABLE TIME
1A81 8C8F A        LD      AC3,ATCTP
1A82 CE06 A        ADD     AC3,D6(AC2)            ; AC3 POINTS TO CI(T)
1A83        ;                                  DETERMINE ADDITIONAL TIME NEEDED
1A83 886F A        LD      AC2,APTEM
1A84 D200 A        SUB     AC0,D0(AC2)
1A85 C300 A        ADD     AC0,D0(AC3)
1A86 1205 A        BOC     CCZP,PM4               ; BRANCH IF DELTA E TIME AVAILABLE
1A87 C200 A        ADD     AC0,D0(AC2)
1A88 4D00 A        LI      AC1,C0                 ; SET DURATION OF CI(T) TO ZERO
1A89 A700 A        ST      AC1,D0(AC3)
1A8A 2103 A        JMP     PM5
1A8B D200 A PM3:   SUB     AC0,D0(AC2)            ; REDUCE CI(P)
1A8C A300 A PM4:   ST      AC0,D0(AC3)
1A8D 906F A        LD      AC0,@APTEM             ; AC0 = DELTA E
1A8E        PM5:
1A8E 8896 A        LD      AC2,ATECI
1A8F 8C8F A        LD      AC3,ATCTP              ; GENERATE POINTER TO EI
1A90 CE02 A        ADD     AC3,D2(AC2)
1A91 3181 A        RCPY    AC0,AC1
1A92 C300 A        ADD     AC0,D0(AC3)            ; EXTEND EI
1A93 A300 A        ST      AC0,D0(AC3)
1A94        ;                                  INCREMENT CURRENT INTERVAL TIMER
1A94 8C85 A        LD      AC3,ATINT
1A95 C700 A        ADD     AC1,D0(AC3)
1A96 B485 A        ST      AC1,@ATINT
1A97 0200 A        RTS     D0                     ; RETURN
```

```
1A98                    .PAGE 'DETECTOR SWITCH DEBOUNCER'
1A98                    .LIST   LPR
1A98            PSWCH:
1A98            ;                                   DEBOUNCE SDU DETECTOR SWITCHES
1A98            ;
1A98            ;                                   CHECKOUT ROUTINE FOR INTERFACE
1A98            ;                                   BETWEEN LAB INTERSECTION MODEL AND
1A98            ;                                   VEHICLE SIGNATURE DATA
1A98            ;
1A98 9068 A             LD      AC0,@ABBUF              ; AC0 = DDIN
1A99 8880 A             LD      AC2,AVCSDB
1A9A            ;                                   DEBOUNCE DOWNSTREAM SWITCH
1A9A 075C A             SKBIT   D12                     ; SKIP IF DOWNSTREAM SWITCH UNACTIVATED
1A9B 2103 A             JMP     .+4
1A9C            ;                                   SWITCH IS UNACTIVATED
1A9C 4D00 A             LI      AC1,C0
1A9D A511 A             ST      AC1,PSWFC               ; CLEAR FLAG
1A9E 2105 A             JMP     .+6                     ; CHECK UPSTREAM DETECTOR
1A9F            ;                                   SWITCH IS ACTIVATED
1A9F 4D01 A             LI      AC1,C1
1AA0 E50E A             SKG     AC1,PSWFC               ; SKIP IF FLAG IS CLEAR
1AA1 2102 A             JMP     .+3
1AA2 A50C A             ST      AC1,PSWFC               ; SET FLAG
1AA3            ;                                   MODIFY NEXT INSTR TO ACCESS OTHER
1AA3            ;                                   SIGNATURES IN VCSDB
1AA3 A60F A             ST      AC1,D15(AC2)            ; SET FOS STATUS WORD FOR DOWNSTREAM
1AA4            ;                                   DEBOUNCE UPSTREAM SWITCH
1AA4 075D A             SKBIT   D13                     ; SKIP IF UPSTREAM SWITCH UNACTIVATED
1AA5 2103 A             JMP     .+4
1AA6            ;                                   SWITCH IS UNACTIVATED
1AA6 4D00 A             LI      AC1,C0
1AA7 A508 A             ST      AC1,PSWFD               ; CLEAR FLAG
1AA8 0200 A             RTS     D0                      ; RETURN TO PMON
1AA9            ;                                   SWITCH IS ACTIVATED
1AA9 4D01 A             LI      AC1,C1
1AAA E505 A             SKG     AC1,PSWFD               ; SKIP IF FLAG IS CLEAR
1AAB 0200 A             RTS     D0
1AAC A503 A             ST      AC1,PSWFD               ; SET FLAG
1AAD            ;                                   MODIFY NEXT INSTR TO ACCESS OTHER
1AAD            ;                                   SIGNATURES IN VCSDB
1AAD A607 A             ST      AC1,D7(AC2)             ; SET FOS STATUS WORD FOR UPSTREAM
1AAE 0200 A             RTS     D0                      ; RETURN
1AAF            ;
1AAF            ;                                   LOCAL DATA
1AAF 0000 A PSWFC:      .WORD   0
1AB0 0000 A PSWFD:      .WORD   0
```

```
1AB1                    .PAGE
1AB1                    .ASECT
1AB1 1C19 A             .=X'1C19
1C19          ;
1C19          ;
```

```
1C19                .LIST   LVC
1C19        VCLAS:
1C19        ;                                   VEHICLE CLASSIFICATION
1C19        ;                                   MAY 30 1975
1C19        ;
1C19 2100 A         JMP     .+1                     ; JSR @ACOTF MAY BE INSERTED HERE
1C1A 9C7E A         LD      AC3,@AVCDT
1C1B        ;                                   IS THE SPECIFIED DETECTOR IN RANGE
1C1B ED66 A         SKG     AC3,VCHD
1C1C ED66 A         SKG     AC3,VCLD
1C1D 2154 A         JMP     VCER1
1C1E        ;                                   LOCATE THIS DETECTOR'S PORTION OF SDB
1C1F 4BFF A         AISZ    AC3,CM1
1C1F 2100 A         JMP     .+1
1C20 5F03 A         SHL     AC3,D3
1C21 8880 A         LD      AC2,AVCSDB
1C22 3800 A         RADD    AC2,AC3
1C23 837F A         LD      AC2,AVCDB
1C24        ;                                   CHECK EOS STATUS REGISTER
1C24 8307 A         LD      AC0,D7(AC3)
1C25 1118 A         BOC     CCEZ,VCRET
1C26        ;
1C26        ;                                   ALL DATA FROM THE VEHICLE SIGNATURE DATA
1C26        ;                                   BASE IS SCALED BO
1C26        ;
1C26        ;                                   COPY VEHICLE SIGNATURE DATA BASE
1C26 8305 A         LD      AC0,D5(AC3)
1C27 A205 A         ST      AC0,D5(AC2)
1C28 8300 A         LD      AC0,D0(AC3)
1C29 A200 A         ST      AC0,D0(AC2)
1C2A 8301 A         LD      AC0,D1(AC3)
1C2B A201 A         ST      AC0,D1(AC2)
1C2C 8302 A         LD      AC0,D2(AC3)
1C2D A202 A         ST      AC0,D2(AC2)
1C2E 8303 A         LD      AC0,D3(AC3)
1C2F A203 A         ST      AC0,D3(AC2)
1C30 8304 A         LD      AC0,D4(AC3)
1C31 A204 A         ST      AC0,D4(AC2)
1C32 8306 A         LD      AC0,D6(AC3)
1C33 A206 A         ST      AC0,D6(AC2)
1C34        ;                                   RECHECK EOS STATUS REGISTER
1C34 8307 A         LD      AC0,D7(AC3)
1C35        ;                                   RESET EOS STATUS REGISTER
1C35 4000 A         LI      AC1,C0
1C36 A707 A         ST      AC1,D7(AC3)
1C37 1508 A         BOC     CCNZ,VC01
1C38 947E A         LD      AC1,@AVCDT
1C39 8C77 A         LD      AC3,APERR
1C3A 2C4F A         JSR     @AERE6
1C3B 8205 A         LD      AC0,D5(AC2)
1C3C 1503 A         BOC     CCNZ,VC01
1C3D 2C50 A         JSR     @AERE7
```

```
1C3E 4C01 A VCRET:  LI      AC0,C1
1C3F 0200 A         RTS     D0                      ; RETURN FROM VCLAS
1C40        ;                                   TEST FOR NOISE
1C40 4C20 A VC01:   LI      AC0,32
1C41 E200 A         SKG     AC0,D0(AC2)
1C42 210A A         JMP     VC02
1C43        ;                                   TOTAL TIME OF SIGNATURE TOO SHORT FOR VEHICLE
1C43        ;                                   SIGNATURE ASSUMED TO BE NOISE
1C43 947E A         LD      AC1,@AVCDT
1C44 8C77 A         LD      AC3,APERR
1C45 2C4E A         JSR     @AERE5
1C46 4C01 A         LI      AC0,C1
1C47 8C7D A         LD      AC3,AVCLOG
1C48 7801 A         ISZ     D1(AC3)
1C49 0200 A         RTS     D0                      ; RETURN FROM VCLAS
1C4A 7800 A         ISZ     D0(AC3)
1C4B 3081 A         NOP
1C4C 0200 A         RTS     D0                      ; RETURN FROM VCLAS
1C4D        ;                                   INCREMENT DETECTOR COUNTER
1C4D        VC02:
1C4D 4FFF A         LI      AC3,CM1
1C4E 947E A         LD      AC1,@AVCDT
1C4F 3700 A         RADD    AC1,AC3
1C50 CC7C A         ADD     AC3,AVCVC
1C51 7800 A         ISZ     D0(AC3)
1C52 2100 A         JMP     .+1
1C53        ;                                   GROSS TEST
1C53 4C00 A         LI      AC0,C0
1C54 F205 A         SKNE    AC0,D5(AC2)
1C55 2115 A         JMP     VCGTF
1C56        ;                                   CHECK MAGNITUDE OF SIGNATURE DATA
1C56 812D A         LD      AC0,VCMA
1C57 7201 A         SKAZ    AC0,D1(AC2)
1C58 211D A         JMP     VCER2
1C59 7203 A         SKAZ    AC0,D3(AC2)
1C5A 211B A         JMP     VCER2
1C5B 7205 A         SKAZ    AC0,D5(AC2)
1C5C 2119 A         JMP     VCER2
1C5D 8127 A         LD      AC0,VCMT
1C5E 7200 A         SKAZ    AC0,D0(AC2)
1C5F 211A A         JMP     VCER3
1C60 7202 A         SKAZ    AC0,D2(AC2)
1C61 2118 A         JMP     VCER3
1C62 7204 A         SKAZ    AC0,D4(AC2)
1C63 2116 A         JMP     VCER3
1C64 7206 A         SKAZ    AC0,D6(AC2)
1C65 2114 A         JMP     VCER3
1C66        ;                                   CHECK FOR TIME OVERFLOW
1C66 811F A         LD      AC0,VCMOF
1C67 6200 A         AND     AC0,D0(AC2)
1C68 F11D A         SKNE    AC0,VCMOF
1C69 2152 A         JMP     VCHFE
```

```
1C6A 2127 A          JMP     VCDHK
1C6B 4C01 A VCGTF:   LI      AC0,C1
1C6C 8C7D A          LD      AC3,AVCLOG
1C6D 7B03 A          ISZ     D3(AC3)
1C6E 0200 A          RTS     D0                    ; RETURN FROM VCLAS
1C6F 7B02 A          ISZ     D2(AC3)
1C70 2100 A          JMP     .+1
1C71 0200 A          RTS     D0                    ; RETURN FROM VCLAS
1C72         VCER1:
1C72         ;                                  DETECTOR SPECIFIED DOES NOT EXIST
1C72 947E A          LD      AC1,@AVCDT
1C73 8C77 A          LD      AC3,APERR
1C74 2C4A A          JSR     @AERE1
1C75 21C8 A          JMP     VCRET
1C76         VCER2:
1C76         ;                                  VEHICLE SIGNATURE DATA EXCEEDS BOUNDS (AMP)
1C76 947E A          LD      AC1,@AVCDT
1C77 8C77 A          LD      AC3,APERR
1C78 2C4B A          JSR     @AERE2
1C79 21C4 A          JMP     VCRET
1C7A         VCER3:
1C7A         ;                                  VEHICLE SIGNATURE DATA EXCEEDS BOUNDS (TIME)
1C7A 947E A          LD      AC1,@AVCDT
1C7B 8C77 A          LD      AC3,APERR
1C7C 2C4C A          JSR     @AERE3
1C7D 21C0 A          JMP     VCRET
1C7E         VCER4:
1C7E         ;                                  VEHICLE TIME DATA OUT OF ORDER
1C7E 8C77 A          LD      AC3,APERR
1C7F 947E A          LD      AC1,@AVCDT
1C80 2C4D A          JSR     @AERE4
1C81 21BC A          JMP     VCRET
```

```
1C82                    .LIST   LVC
1C82            ;                                   DETECTOR INDEX RANGE
1C82 0001 A VCHD:       .WORD   DETS                    ; HIGHEST DETECTOR INDEX NUMBER EXPECTED
1C83 0000 A VCLD:       .WORD   0                       ; LOWEST DETECTOR INDEX NUMBER - 1
1C84            ;                                   MASKS
1C84 FF00 A VCMA:       .WORD   X'FF00                  ; MASK USED IN AMPLITUDE IN RANGE TEST
1C85 F000 A VCMT:       .WORD   X'F000                  ; MASK USED IN TIME IN RANGE TEST
1C86 0FFF A VCMOF:      .WORD   X'0FFF                  ; MASK USED IN TIME OVERFLOW TEST
1C87            ;                                   THE FOLLOWING ARE USED IN VCDHK
1C87 0028 A VCST1:      .WORD   40                      ; 2.50 B4       USED IN L2 TEST
1C88 0007 A VCST2:      .WORD   7                       ; 0.43 B4       USED IN L1 TEST
1C89 000B A VCST3:      .WORD   11                      ; 0.70 B4       USED IN L1 TEST
1C8A            ;                                   THE FOLLOWING ARE USED IN VCHFE
1C8A 2BAE A VC31:       .WORD   11182                   ; 2.730 B13     G1 COEFFICIENT FOR H1
1C8B 0E04 A VC32:       .WORD   3588                    ; 0.876 B13     G2 COEFFICIENT FOR H1
1C8C 2851 A VC41:       .WORD   10321                   ; 2.520 B13     G1 COEFFICIENT FOR H2
1C8D 0239 A VC42:       .WORD   569                     ; 0.139 B13     G2 COEFFICIENT FOR H2
1C8E 3EE1 A H1T1:       .WORD   16097                   ; 3.93 B13      H1 TEST LOWER BOUND
1C8F 6785 A H1T2:       .WORD   26501                   ; 6.47 B13      H1 TEST UPPER BOUND
1C90 3000 A H2T1:       .WORD   12288                   ; 3.00 B13      H2 TEST LOWER BOUND
1C91 5000 A H2T2:       .WORD   20480                   ; 5.00 B13      H2 TEST UPPER BOUND
```

```
3733 1C92                .LIST   LVC
3734 1C92        VCDHK:
3735 1C92        ;                                  VEHICLE CLASSIFICATION  MOVING/STOPPED TEST
3736 1C92        ;
3737 1C92        ;                                  INSURE TIME DATA WORDS IN PROPER ORDER
3738 1C92 8204 A         LD      AC0,D4(AC2)
3739 1C93 E202 A         SKG     AC0,D2(AC2)
3740 1C94 21E9 A         JMP     VCER4
3741 1C95 8206 A         LD      AC0,D6(AC2)
3742 1C96 E204 A         SKG     AC0,D4(AC2)
3743 1C97 21E6 A         JMP     VCER4
3744 1C98 8200 A         LD      AC0,D0(AC2)
3745 1C99 E206 A         SKG     AC0,D6(AC2)
3746 1C9A 21E3 A         JMP     VCER4
3747 1C9B        ;                                  FORM T(C1) + T(C4) - T(C3)
3748 1C9B 8200 A         LD      AC0,D0(AC2)             ; AC0 = T(C4)
3749 1C9C D206 A         SUB     AC0,D6(AC2)             ; AC0 = T(C4) - T(C3)
3750 1C9D C202 A         ADD     AC0,D2(AC2)             ; AC0 = T(C4) - T(C3) + T(C1)
3751 1C9E A207 A         ST      AC0,D7(AC2)             ; R0
3752 1C9F        ;                                  FORM F0
3753 1C9F 8602 A         LD      AC1,D2(AC2)             ; R0
3754 1CA0 5D04 A         SHL     AC1,D4                  ; R4
3755 1CA1 4C00 A         LI      AC0,C0
3756 1CA2 C172 A         ADD     AC0,VCC0                ; CLEAR OVERFLOW STATUS BIT
3757 1CA3 0690 A         DIV     D7(AC2)                 ; R4
     1CA4 0007 A
3758 1CA5 2100 A         JMP     .+1
3759 1CA6        ;                                  CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
3760 1CA6        ;                                  JUMP TO VCHFE
3761 1CA6 074E A         SKSTF   D14
3762 1CA7 2101 A         JMP     .+2
3763 1CA8 2113 A         JMP     VCHFE
3764 1CA9        ;                                  AC1 = T(C1) / T(C1) + T(C4) - T(C3)
3765 1CA9        ;                                  VEHICLE IS CONSIDERED MOVING VEHICLE IF
3766 1CA9        ;                                  0.43 < F0 < 0.7
3767 1CA9 E5DE A         SKG     AC1,VCST2               ; SKIP IF AC1 > 0.43
3768 1CAA 2111 A         JMP     VCHFE
3769 1CAB E5DD A         SKG     AC1,VCST3               ; SKIP IF AC1 > 0.7
3770 1CAC F5DC A         SKNE    AC1,VCST3               ; SKIP IF AC1 NOT = 0.7
3771 1CAD 210E A         JMP     VCHFE                   ; JUMP TO STOPPED VEHICLE CLASSIFIER
3772 1CAE        ;                                  FORM F1 + F2
3773 1CAE 8606 A         LD      AC1,D6(AC2)             ; R0
3774 1CAF D602 A         SUB     AC1,D2(AC2)             ; AC1 = T(C3) - T(C1)
3775 1CB0 5D04 A         SHL     AC1,D4                  ; R4
3776 1CB1 4C00 A         LI      AC0,C0
3777 1CB2 C162 A         ADD     AC0,VCC0                ; CLEAR OVERFLOW STATUS BIT
3778 1CB3 0690 A         DIV     D7(AC2)                 ; R4
     1CB4 0007 A
3779 1CB5 2100 A         JMP     .+1
3780 1CB6        ;                                  CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
3781 1CB6        ;                                  JUMP TO VCHFE
3782 1CB6 074E A         SKSTF   D14
```

```
3783 1CB7 2101 A        JMP     .+2
3784 1CB8 2103 A        JMP     VCHFE
3785 1CB9        ;                          AC1 = T(C3) - T(C1) / T(C1) + T(C4) - T(C3)
3786 1CB9        ;                          VEHICLE IS CONSIDERED STOPPED VEHICLE IF
3787 1CB9        ;                          F1 + F2 > 2.5
3788 1CB9 E5CD A        SKG     AC1.VCST1          ; SKIP IF AC1 > 2.5
3789 1CBA 2132 A        JMP     VCKFE              ; JUMP TO MOVING VEHICLE CLASSIFIER
3790 1CBB 2100 A        JMP     VCHFE
```

```
1CBC                    .LIST   LVC
1CBC          ;                                   VEHICLE CLASSIFICATION
1CBC          ;                                   STOPPED VEHICLE CLASSIFIER
1CBC        VCHFE:
1CBC 8C7D A           LD      AC3,AVCLOG
1CBD 7B05 A           ISZ     D5(AC3)
1CBE 2102 A           JMP     .+3
1CBF 7B04 A           ISZ     D4(AC3)
1CC0 2100 A           JMP     .+1
1CC1          ;                                   INCREMENT STOPPED VEHICLE COUNTER
1CC1          ;                                   FORM C31 * A1
1CC1 85C8 A           LD      AC1,VC31                ; B13
1CC2 0680 A           MPY     D3(AC2)
1CC3 0003 A
1CC4 A208 A           ST      AC0,D8(AC2)
1CC5 A609 A           ST      AC1,D9(AC2)
1CC6          ;                                   FORM C32 * A2
1CC6 85C4 A           LD      AC1,VC32                ; B13
1CC7 0680 A           MPY     D5(AC2)
1CC8 0005 A
1CC9          ;                                   FORM (C31*A1 + C32*A2) / A0
1CC9 06A0 A           DADD    D8(AC2)                 ; B13
1CCA 0008 A
1CCB C149 A           ADD     AC0,VCC0                ; CLEAR OVERFLOW STATUS BIT
1CCC 0690 A           DIV     D1(AC2)                 ; B13
1CCD 0001 A
1CCE 2100 A           JMP     .+1
1CCF          ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1CCF          ;                                   JUMP TO VCNB5
1CCF 074E A           SKSTF   D14
1CD0 2101 A           JMP     .+2
1CD1 2552 A           JMP     @AVCVNB
1CD2          ;                                   IF VEHICLE IS NOT A BUS, JUMP TO VCVNB.
1CD2 E5BB A           SKG     AC1,H1T1                ; SKIP IF AC1 > 3.93
1CD3 2550 A           JMP     @AVCVNB
1CD4 E5B4 A           SKG     AC1,H1T2                ; SKIP IF AC1 => 6.47
1CD5 2101 A           JMP     .+2
1CD6 254D A           JMP     @AVCVNB
1CD7          ;                                   FORM C41 * A1
1CD7 85B4 A           LD      AC1,VC41                ; B13
1CD8 0680 A           MPY     D3(AC2)
1CD9 0003 A
1CDA A208 A           ST      AC0,D8(AC2)
1CDB A609 A           ST      AC1,D9(AC2)
1CDC          ;                                   FORM C42 * A2
1CDC 85B0 A           LD      AC1,VC42                ; B13
1CDD 0680 A           MPY     D5(AC2)
1CDE 0005 A
1CDF          ;                                   FORM (C41*A1 + C42*A2) / A0
1CDF 06A0 A           DADD    D8(AC2)                 ; B13
1CE0 0008 A
1CE1 C133 A           ADD     AC0,VCC0                ; CLEAR OVERFLOW STATUS BIT
```

```
 3838 1CE2 0690 A          DIV     D1(AC2)              ; B13
      1CE3 0001 A
 3839 1CE4 2100 A          JMP     .+1
 3840 1CE5         ;                                    CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
 3841 1CE5         ;                                    JUMP TO VCNBS
 3842 1CE5 074E A          SKSTF   D14
 3843 1CE6 2101 A          JMP     .+2
 3844 1CE7 253C A          JMP     @AVCVNB
 3845 1CE8         ;                                    IF VEHICLE IS NOT A BUS, JUMP TO VCVNB
 3846 1CE8 E5A7 A          SKG     AC1,H2T1             ; SKIP IF AC1 > 3.0
 3847 1CE9 253A A          JMP     @AVCVNB
 3848 1CEA E5A6 A          SKG     AC1,H2T2             ; SKIP IF AC1 > 5.0
 3849 1CEB 2537 A          JMP     @AVCBUS
 3850 1CEC 2537 A VCHE:    JMP     @AVCVNB
```

```
1CFD                    .LIST   LVC
1CFD            ;                                   VEHICLE CLASSIFICATION
1CFD            ;                                   MOVING VEHICLE CLASSIFIER
1CFD            ;                                       BIT 13 = 1.0
1CFD            VCKFE:
1CFD 8C7D A             LD      AC3,AVCLOG
1CFE 7807 A             ISZ     D7(AC3)
1CFF 2102 A             JMP     .+3
1CF0 7806 A             ISZ     D6(AC3)
1CF1 2100 A             JMP     .+1
1CF2 8021 A             LD      AC3,VCCP
1CF3 4C00 A             LI      AC0,C0
1CF4 A20C A             ST      AC0,D12(AC2)
1CF5 A20F A             ST      AC0,D15(AC2)
1CF6            VCK:
1CF6            ;                                   FORM C4*A2 = G2'
1CF6 8704 A             LD      AC1,D4(AC3)             ; B13
1CF7 0680 A             MPY     D5(AC2)
1CF8 0005 A
1CF9 A208 A             ST      AC0,D8(AC2)
1CFA A609 A             ST      AC1,D9(AC2)
1CFB            ;                                   FORM C3*A1 = G1'
1CFB 8703 A             LD      AC1,D3(AC3)             ; B13
1CFC 0680 A             MPY     D3(AC2)
1CFD 0003 A
1CFE A20A A             ST      AC0,D10(AC2)
1CFF A60B A             ST      AC1,D11(AC2)
1D00            ;                                   FORM ABS(G1' - G2') / A0
1D00 F208 A             SKNE    AC0,D8(AC2)             ; SKIP IF UPPER G1' NOT = UPPER G2'
1D01 2103 A             JMP     .+4
1D02 E205 A             SKG     AC0,D8(AC2)             ; SKIP IF UPPER G1' > UPPER G2'
1D03 212F A             JMP     VCK3
1D04 2122 A             JMP     VCK21
1D05 651C A             AND     AC1,VCM15
1D06 F51B A             SKNE    AC1,VCM15               ; SKIP IF BIT 15 IS NOT SET
1D07 2108 A             JMP     VCK1
1D08 8519 A             LD      AC1,VCM15
1D09 6609 A             AND     AC1,D9(AC2)
1D0A F517 A             SKNE    AC1,VCM15               ; SKIP IF BIT 15 IS NOT SET
1D0B 2127 A             JMP     VCK3
1D0C 860B A VCK0:       LD      AC1,D11(AC2)
1D0D E509 A             SKG     AC1,D9(AC2)             ; SKIP IF LOWER G1' > LOWER G2'
1D0E 2124 A             JMP     VCK3
1D0F 2116 A             JMP     VCK2
1D10 6609 A VCK1:       AND     AC1,D9(AC2)
1D11 F510 A             SKNE    AC1,VCM15
1D12 21F9 A             JMP     VCK0
1D13 2112 A             JMP     VCK2
```

```
1D14                    .LIST   LVC
1D14 1D15 A VCCP:       .WORD   VCCO
1D15                ;                                   THE FOLLOWING IS USED TO CLEAR THE OVERFLOW
1D15                ;                                   STATUS BIT BEFORE EACH DIVIDE OPERATION
1D15 0000 A VCCO:       .WORD   0
1D16                ;                                   THE FOLLOWING ARE USED IN VCKFE
1D16 046E A             .WORD   1134                        ; 0.277 B13     F1 COEFFICIENT FOR K4
1D17 0051 A             .WORD   81                          ; 0.0198 B13    F2 COEFFICIENT FOR K4
1D18 03A5 A             .WORD   933                         ; 0.228 B13     G1 COEFFICIENT FOR K4
1D19 0F26 A             .WORD   3878                        ; 0.947 B13     G2 COEFFICIENT FOR K4
1D1A 0000 A             .WORD   0
1D1B 0F5C A             .WORD   3932                        ; 0.96 B13      K4 BIAS
1D1C                ;
1D1C 1A8F A             .WORD   6799                        ; 1.66 B13      F1 COEFFICIENT FOR K5
1D1D 15C2 A             .WORD   5570                        ; 1.36 B13      F2 COEFFICIENT FOR K5
1D1E 22B8 A             .WORD   8888                        ; 2.17 B13      G1 COEFFICIENT FOR K5
1D1F 4199 A             .WORD   16793                       ; 4.10 B13      G2 COEFFICIENT FOR K5
1D20 0000 A             .WORD   0
1D21 171B A             .WORD   5915                        ; 1.444 B13     K5 BIAS
1D22                ;                                   MASK
1D22 8000 A VCM15:      .WORD   X'8000                      ; MASK FOR DOUBLE PRECISION COMPARISIONS
1D23 1D92 A AVCBUS:     .WORD   VCBUS                       ; POINTER TO VEHICLE IS BUS ROUTINE
1D24 1D9B A AVCVNB:     .WORD   VCVNB                       ; POINTER TO VEHICLE NOT BUS ROUTINE
1D25 1DA2 A AVCER:      .WORD   VCER                        ; POINTER TO OVERFLOW ERROR ROUTINE
```

```
3927 1D26                .LIST   LVC
3928 1D26        ;                                   G1' > G2'
3929 1D26 860B A VCK2:    LD      AC1,D11(AC2)
3930 1D27 06B0 A VCK21:   DSUB    D8(AC2)                ; B13     FORM G1' - G2'
     1D28 0008 A
3931 1D29 C1EB A          ADD     AC0,VCC0               ; CLEAR OVERFLOW STATUS BIT
3932 1D2A 0690 A          DIV     D1(AC2)                ; FORM (G1' - G2') / A0
     1D2B 0001 A
3933 1D2C        ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
3934 1D2C        ;                                   JUMP TO VCER
3935 1D2C 074E A          SKSTF   D14
3936 1D2D 2101 A          JMP     .+2
3937 1D2E 25F6 A          JMP     @AVCER
3938 1D2F A60D A          ST      AC1,D13(AC2)
3939 1D30 4C00 A          LI      AC0,C0
3940 1D31 A20C A          ST      AC0,D12(AC2)
3941 1D32 210F A          JMP     VCK5
3942 1D33        ;                                   G1' < G2'
3943 1D33 820B A VCK3:    LD      AC0,D8(AC2)            ; B13
3944 1D34 8609 A          LD      AC1,D9(AC2)
3945 1D35 06B0 A          DSUB    D10(AC2)               ; FORM (G2' - G1') / A0
     1D36 000A A
3946 1D37 C1DD A          ADD     AC0,VCC0               ; CLEAR OVERFLOW STATUS BIT
3947 1D38 0690 A          DIV     D1(AC2)                ; FORM (G2' - G1') / A0
     1D39 0001 A
3948 1D3A        ;                                   CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
3949 1D3A        ;                                   JUMP TO VCER
3950 1D3A 074E A          SKSTF   D14
3951 1D3B 2101 A          JMP     .+2
3952 1D3C 25E8 A          JMP     @AVCER
3953 1D3D        ;                                   CONVERT QUOTIENT TO NEGATIVE AND STORE
3954 1D3D 5101 A          CAI     AC1,C1
3955 1D3E A60D A          ST      AC1,D13(AC2)
3956 1D3F 4C00 A          LI      AC0,C0
3957 1D40 5000 A          CAI     AC0,C0
3958 1D41 A20C A          ST      AC0,D12(AC2)
3959 1D42        ;                                   FORM C11*(T(C2) - T(C1))
3960 1D42 8604 A VCK5:    LD      AC1,D4(AC2)            ; B0
3961 1D43 D602 A          SUB     AC1,D2(AC2)
3962 1D44 0780 A          MPY     D1(AC3)                ; B13
     1D45 0001 A
3963 1D46 A208 A          ST      AC0,D8(AC2)
3964 1D47 A609 A          ST      AC1,D9(AC2)
3965 1D48        ;                                   FORM C12 * (T(C3) - T(C2))  = F2'
3966 1D48 8606 A          LD      AC1,D6(AC2)
3967 1D49 D604 A          SUB     AC1,D4(AC2)
3968 1D4A 0780 A          MPY     D2(AC3)
     1D4B 0002 A
3969 1D4C A20A A          ST      AC0,D10(AC2)
3970 1D4D A60B A          ST      AC1,D11(AC2)
3971 1D4E        ;                                   FORM ABS(F2' - F1')
3972 1D4E F208 A          SKNE    AC0,D8(AC2)            ; SKIP IF UPPER F2' NOT = UPPER F1'
```

```
1D4F 2103 A          JMP     .+4
1D50 E208 A          SKG     AC0,D8(AC2)          ; SKIP IF UPPER F2' > UPPER F1'
1D51 2125 A          JMP     VCK9
1D52 210E A          JMP     VCK8
1D53 65CE A          AND     AC1,VCM15
1D54 F5CD A          SKNE    AC1,VCM15            ; SKIP IF BIT 15 NOT SET
1D55 2108 A          JMP     VCK7
1D56 85CB A          LD      AC1,VCM15
1D57 6609 A          AND     AC1,D9(AC2)
1D58 F5C9 A          SKNE    AC1,VCM15            ; SKIP IF BIT 15 NOT SET
1D59 211D A          JMP     VCK9
1D5A 860B A VCK6:    LD      AC1,D11(AC2)
1D5B E609 A          SKG     AC1,D9(AC2)          ; SKIP IF LOWER F2' > LOWER F1'
1D5C 211A A          JMP     VCK9
1D5D 2103 A          JMP     VCK8
1D5E 6609 A VCK7:    AND     AC1,D9(AC2)
1D5F F5C2 A          SKNE    AC1,VCM15
1D60 21F9 A          JMP     VCK6
1D61        ;                                 F2' > F1'
1D61 4000 A VCK8:    LI      AC1,C0               ; B0
1D62 F60C A          SKNE    AC1,D12(AC2)         ; SKIP IF G2' - G1' NEGATIVE
1D63 2137 A          JMP     VCVNB                ; OTHERWISE JUMP TO NON-BUS ROUTINE
1D64 820A A          LD      AC0,D10(AC2)         ; B13
1D65 860B A          LD      AC1,D11(AC2)
1D66 06B0 A          DSUB    D8(AC2)
1D67 0008 A
1D68 C1AC A          ADD     AC0,VCC0             ; CLEAR OVERFLOW STATUS BIT
1D69 0690 A          DIV     D7(AC2)
1D6A 0007 A
1D6B        ;                                 CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
1D6B        ;                                 JUMP TO VCER
1D6B 074E A          SKSTF   D14
1D6C 2101 A          JMP     .+2
1D6D 25B7 A          JMP     @AVCER
1D6E 4C00 A          LI      AC0,C0
1D6F 06A0 A          DADD    D12(AC2)
1D70 000C A
1D71 3081 A          NOP
1D72 07A0 A          DADD    D5(AC3)
1D73 0005 A
1D74 2100 A          JMP     .+1
1D75 1225 A          BOC     CCZP,VCVNB           ; IF RESULTS =>0, JUMP TO NON-BUS ROUTINE
1D76 2114 A          JMP     VCNK
1D77        ;                                 F2' < F1'
1D77 8208 A VCK9:    LD      AC0,D8(AC2)          ; B13
1D78 8609 A          LD      AC1,D9(AC2)
1D79 06B0 A          DSUB    D10(AC2)
1D7A 000A A
1D7B C199 A          ADD     AC0,VCC0             ; CLEAR OVERFLOW STATUS BIT
1D7C 0690 A          DIV     D7(AC2)
1D7D 0007 A
1D7E        ;                                 CHECK FOR OVERFLOW. IF OVERFLOW OCCURS
```

```
1D7E          ;                                   JUMP TO VCER
1D7E 074E A          SKSTF   D14
1D7F 2101 A          JMP     .+2
1D80 25A4 A          JMP     @AVCER
1D81 A61C A          ST      AC1,D16(AC2)
1D82 4C00 A          LI      AC0,C0
1D83 8706 A          LD      AC1,D6(AC3)
1D84 06B0 A          DSUB    D15(AC2)
1D85 000F A
1D86 3081 A          NOP
1D87 06A0 A          DADD    D12(AC2)
1D88 000C A
1D89 2100 A          JMP     .+1
1D8A 1210 A          BOC     CCZP,VCVNB          ; IF RESULTS =>0, JUMP TO NON-BUS ROUTINE
1D8B          VCNK:
1D8B FD88 A          SKNE    AC3,VCCP            ; B0
1D8C 2101 A          JMP     .+2
1D8D 2104 A          JMP     VCBUS
1D8E 4C06 A          LI      AC0,C6
1D8F 3300 A          RADD    AC0,AC3
1D90 2500 A          JMP     @AVCK
1D91 1CF6 A AVCK:    .WORD   VCK
```

```
1D92                    .LIST   LVC
1D92 4C00 A VCBUS:      LI      AC0,C0
1D93 4D19 A             LI      AC1,DPMD            ; AC0 = PREEMPT MINIMUM DURATION
1D94 B47B A             ST      AC1,@AVCBT
1D95 8C7D A             LD      AC3,AVCLOG
1D96 7B09 A             ISZ     D9(AC3)
1D97 0200 A             RTS     D0
1D98 7B08 A             ISZ     D8(AC3)
1D99 2100 A             JMP     .+1
1D9A 0200 A             RTS     D0
1D9B 4C01 A VCVNB:      LI      AC0,C1
1D9C 8C7D A             LD      AC3,AVCLOG
1D9D 7B0B A             ISZ     D11(AC3)
1D9E 0200 A             RTS     D0
1D9F 7B0A A             ISZ     D10(AC3)
1DA0 2100 A             JMP     .+1
1DA1 0200 A             RTS     D0
1DA2            VCER:
1DA2            ;                           ERROR DETECTED   OVERFLOW ON DIVIDE
1DA2 947E A             LD      AC1,@AVCDT
1DA3 8C77 A             LD      AC3,APERR
1DA4 2C49 A             JSR     @AEREO
1DA5 4C01 A             LI      AC0,C1
1DA6 0200 A             RTS     D0
```

```
4065 1DA7                .PAGE
4066 1DA7                .ASECT
4067 1DA7 1E00 A         .=X'1E00
```

```
1E00                    .LIST   LEB
1E00          ;
1E00          ;                    ... BACKGROUND PROGRAMS ...
1E00          EBKD:
1E00          ;
1E00          ;                    ... RAM ADDRESS TEST ...
1E00 80D6 A   EBKD1:  LD      AC0,ADEND        ; START ADDRESS
1E01 80D4 A           ST      AC0,@ASSAD
1E02 80F6 A           LD      AC0,H3BFF        ; END ADDRESS
1E03 80D5 A           ST      AC0,@ASEAD
1E04          ;
1E04 2C5A A           JSR     @ASMAD
1E05 2104 A           JMP     EBKDE            ; ERROR RETURN, (AC2) = CONTENTS
1E06                                           ;               (AC3) = ADDRESS
1E06 2100 A           JMP     .+1              ; NO ERROR RETURN
1E07          ;
1E07 0744 A           SKSTF   D4               ; SKIP IF STATUS IS SET
1E08 21F7 A           JMP     EBKD1            ; RESTART RAM ADDRESS TEST
1E09 2100 A           JMP     EBKDE            ; JUMP IF ERROR
1E0A          ;
1E0A          ;                    ... ERROR, TERMINAL WAIT LOOP ...
1E0A 2100 A   EBKDE:  JMP     .+1              ; WAIT FOR MONITOR ROUTINE TO INITIATE
1E0B 21FE A           JMP     .-1              ;    ERROR RESPONSE
```

```
1E0C                    .LIST   LEB
1E0C            ;
1E0C            ;                  ... RAM ADDRESS TEST ...
1E0C 4CFF A SMAD:   LI      AC0,CM1         ; CLEAR ERROR COUNTER
1E0D 88D5 A         LD      AC2,ASEAD       ; POINTER TO END ADDRESS
1E0E 9CD4 A         LD      AC3,@ASSAD      ; START ADDRESS
1E0F            ;
1E0F AF00 A SMAD1:  ST      AC3,(AC3)       ; STORE ADDRESS AT EACH LOCATION
1E10 4B01 A         AISZ    AC3,C1          ; INCREMENT ADDRESS
1E11 EE00 A         SKG     AC3,(AC2)       ; SKIP IF DONE
1E12 21FC A         JMP     SMAD1
1E13            ;
1E13 9CD4 A         LD      AC3,@ASSAD      ; START ADDRESS
1E14 FF00 A SMAD2:  SKNE    AC3,(AC3)       ; SKIP IF MISMATCH
1E15 2106 A         JMP     SMAD3           ; MATCH, CONTENTS = ADDRESS
1E16 4801 A         AISZ    AC0,C1          ; SKIP IF FIRST ERROR
1E17 1301 A         BOC     CCB0,SMADE      ; BRANCH IF SECOND ERROR
1E18 21FB A         JMP     SMAD2           ; CONTINUE LOOP
1E19            ;
1E19            ;                  ... TWO CONSECUTIVE ADDRESS ERRORS ...
1E19 8B00 A SMADE:  LD      AC2,(AC3)       ; GET ERRONEOUS CONTENTS
1E1A 0704 A         SETST   D4              ; SET STATUS FLAG 4
1E1B 0200 A         RTS     D0              ; RETURN FROM SMAD (ERROR)
1E1C            ;
1E1C 4CFF A SMAD3:  LI      AC0,CM1         ; CLEAR ERROR COUNTER
1E1D 4B01 A         AISZ    AC3,C1          ; INCREMENT ADDRESS
1E1E EE00 A         SKG     AC3,(AC2)       ; SKIP IF DONE
1E1F 21F4 A         JMP     SMAD2           ; CONTINUE LOOP
1E20 0201 A         RTS     D1              ; RETURN FROM SMAD (NO ERROR)
```

```
4124 1E21                .PAGE
4125 1E21                .ASECT
4126 1E21 1F00 A         .=X'1F00
4127 1F00        ;
```

```
1F00                    .LIST   LBX
1F00          ;                    ... D A T A   I N P U T   C P C ...
1F00          ;
1F00          ;                    ... DATA INPUT FOR RTC SEQUENCE ...
1F00          ;
1F00 8CE3 A BRTC:   LD      AC3,ATCP        ; INPUT AND SAVE DISCRETE INPUT WORD
1F01 8868 A         LD      AC2,ABBUF
1F02 0403 A         RIN     FC3
1F03 A200 A         ST      AC0,D0(AC2)
1F04 8CE0 A         LD      AC3,ALDU        ; INPUT AND SAVE COMMAND REGISTERS
1F05 0403 A         RIN     FC3
1F06 A201 A         ST      AC0,D1(AC2)
1F07 0404 A         RIN     FC4
1F08 A202 A         ST      AC0,D2(AC2)
1F09          ;
1F09 8CE9 A         LD      AC3,APPDU       ; INPUT AND SAVE PROG. PANEL DISCRETES
1F0A 0403 A         RIN     FC3
1F0B B08B A         ST      AC0,@ABPPD
1F0C 0200 A         PTS     D0
1F0D          ;
1F0D          ;
```

```
1F0D                    .LIST   LBX
1F0D          ;
1F0D          ;                 ... DATA INPUT FROM PROG. PANEL ...
1F0D          ;                     (THUMBWHEEL SWITCH DATA ONLY)
1F0D          ;
1F0D          ;                 ... THIS ROUTINE IS EXECUTED DURING RESTART
1F0D          ;                        INITIALIZATION AND THEREAFTER ONLY UPON INPUT
1F0D          ;                        REQUEST VIA THE PROG PANEL PUSHBUTTON ...
1F0D          ;
1F0D 88BB A BPAN:   LD      AC2,ABPPD       ; BASE ADDRESS OF INPUT DATA BUFFER
1F0E 8CEA A         LD      AC3,APP1U       ; PLAN 1 PERIPHERAL DEVICE ADDRESS
1F0F 2909 A         JSR     BPIN            ; READ AND SAVE PARAMETER SWITCH DATA
1F10          ;
1F10 4A07 A         AISZ    AC2,C7          ; INCREMENT BY LENGTH OF BUFFER
1F11 2100 A         JMP     .+1
1F12 8CEB A         LD      AC3,APP2U       ; PLAN 2 PERIPHERAL DEVICE ADDRESS
1F13 2905 A         JSR     BPIN            ; READ AND SAVE PARAMETER SWITCH DATA
1F14          ;
1F14 4A07 A         AISZ    AC2,C7          ; INCREMENT BY LENGTH OF BUFFER
1F15 2100 A         JMP     .+1
1F16 8CEC A         LD      AC3,APP3U       ; PLAN 3 PERIPHERAL DEVICE ADDRESS
1F17 2901 A         JSR     BPIN            ; READ AND SAVE PARAMETER SWITCH DATA
1F18          ;
1F18 0200 A         RTS     D0              ; RETURN FROM BPAN
1F19          ;
1F19          ;                 ... INPUT A SET OF PROG. PANEL PARAMETER SWITCHES ...
1F19          ;                        (AC2) = BASE ADDRESS OF DATA BUFFER
1F19          ;                        (AC3) = PERIPHERAL DEVICE ADDRESS
1F19 0411 A BPIN:   RIN     FC11
1F1A A201 A         ST      AC0,D1(AC2)
1F1B 0412 A         RIN     FC12
1F1C A202 A         ST      AC0,D2(AC2)
1F1D 0413 A         RIN     FC13
1F1E A203 A         ST      AC0,D3(AC2)
1F1F 0414 A         RIN     FC14
1F20 A204 A         ST      AC0,D4(AC2)
1F21 0415 A         RIN     FC15
1F22 A205 A         ST      AC0,D5(AC2)
1F23 0410 A         RIN     FC10
1F24 A206 A         ST      AC0,D6(AC2)
1F25 0417 A         RIN     FC17
1F26 A207 A         ST      AC0,D7(AC2)
1F27          ;
1F27 0200 A         RTS     D0              ; RETURN FROM BPIN
1F28          ;
```

```
1F28                .LIST   LBX
1F28          ;                 ... D A T A   O U T P U T   C P C ...
1F28          ;                 ... DATA OUTPUT FOR RTC SEQUENCE ...
1F28 8CE0 A XRTC:   LD      AC3,ALDU
1F29 886B A         LD      AC2,AXBUF
1F2A 8201 A         LD      AC0,D1(AC2)
1F2B 0601 A         ROUT    FC1             ; OUTPUT COMMAND WORD 1
1F2C 8202 A         LD      AC0,D2(AC2)
1F2D 0602 A         ROUT    FC2             ; OUTPUT COMMAND WORD 2
1F2E 8CE4 A         LD      AC3,ASDU
1F2F 8200 A         LD      AC0,D0(AC2)
1F30 0601 A         ROUT    FC1             ; OUTPUT STATUS DISPLAY WORD
1F31 0200 A         RTS     D0
1F32          ;
1F32 8CE0 A XWDT:   LD      AC3,ALDU
1F33 4C00 A         LI      AC0,C0
1F34 0641 A         ROUT    FC41            ; RESET CONTROL PULSE FOR DWELL TIMER
1F35 0200 A         RTS     D0
1F36          ;
```

```
1F36                .LIST   LBX
1F36        ;
1F36        ;                   ...FORMAT OUTPUT COMMAND WORDS ...
1F36 8881 A XSDC:   LD      AC2,ATBUF
1F37 8200 A         LD      AC0,D0(AC2)     ; FETCH COMMAND INDEX WORD
1F38 1512 A         BOC     CCNZ,XFOW       ; TEST FOR NON-ZERO WORD FOR NEW COMMAND
1F39 90A0 A         LD      AC0,@AECMF      ; COMMAND FEEDBACK ERROR FLAG
1F3A 1508 A         BOC     CCNZ,XCER       ; BRANCH IF ERROR
1F3B        ;                   ... (INSERT ADDITIONAL TESTS HERE)
1F3B 886B A         LD      AC2,AXBUF
1F3C 8201 A         LD      AC0,D1(AC2)
1F3D 60F0 A         AND     AC0,H7FFF       ; CLEAR UPDATE BIT IN COMMAND WORD 1
1F3E A201 A         ST      AC0,D1(AC2)
1F3F 8202 A         LD      AC0,D2(AC2)
1F40 60F0 A         AND     AC0,H7FFF       ; CLEAR UPDATE BIT IN COMMAND WORD 2
1F41 A202 A         ST      AC0,D2(AC2)
1F42 0200 A         RTS     D0
1F43        ;                   ... COMMAND ERROR, SET UPDATE BIT FOR NEXT OUTPUT ...
1F43 886B A XCER:   LD      AC2,AXBUF
1F44 8201 A         LD      AC0,D1(AC2)     ; XOCW1
1F45 60F0 A         AND     AC0,H7FFF       ; CLEAR UPDATE BIT
1F46 A201 A         ST      AC0,D1(AC2)
1F47 8202 A         LD      AC0,D2(AC2)     ; XOCW2
1F48 072F A         SETBIT  D15             ; SET UPDATE BIT
1F49 A202 A         ST      AC0,D2(AC2)
1F4A 0200 A         RTS     D0
1F4B        ;
1F4B        ;                   ... FORMAT NEW OUTPUT COMMAND WORDS ...
1F4B 6137 A XFOW:   AND     AC0,H7F00       ; EXTRACT MAIN STREET COMMAND INDEX
1F4C 5CF8 A         SHR     AC0,D8          ; SHIFT TO LOWER BITS
1F4D 1102 A         BOC     CCEZ,XMCE
1F4E E147 A         SKG     AC0,XIMX        ; TEST FOR VALID INDEX (1 TO XIMX)
1F4F 210C A         JMP     XFO1
1F50        ;
1F50        ;                   ... INVALID MAIN STREET COMMAND INDEX ...
1F50 8881 A XMCE:   LD      AC2,ATBUF
1F51 8200 A         LD      AC0,D0(AC2)     ; FETCH COMMAND INDEX WORD
1F52 886C A         LD      AC2,AXERR
1F53 A202 A         ST      AC0,D2(AC2)     ; STORE IN ERROR BUFFER, XERRO
1F54 8201 A         LD      AC0,D1(AC2)     ; FETCH ERROR COUNTER AND INCREMENT
1F55 4801 A         AISZ    AC0,C1
1F56 2100 A         JMP     .+1
1F57 A201 A         ST      AC0,D1(AC2)
1F58 8200 A         LD      AC0,D0(AC2)
1F59 0720 A         SETBIT  D0              ; SET ERROR FLAG BIT 0
1F5A A200 A         ST      AC0,D0(AC2)
1F5B 0200 A         RTS     D0
1F5C        ;
1F5C 8D27 A XFO1:   LD      AC3,AXCFT       ; FETCH ADDRESS OF COMMAND FORMAT TABLE
1F5D 3300 A         RADD    AC0,AC3         ; ADD COMMAND INDEX
1F5E 8300 A         LD      AC0,(AC3)       ; FETCH COMMAND FORMAT
1F5F 8C6B A         LD      AC3,AXBUF
```

```
1F60 A301 A          ST      AC0,D1(AC3)      ; STORE IN XOCW1
1F61          ;
1F61 8200 A          LD      AC0,D0(AC2)      ; FETCH COMMAND INDEX WORD
1F62 611F A          AND     AC0,H007F        ; EXTRACT CROSS STREET INDEX
1F63 1102 A          BOC     CCEZ,XCCE
1F64 E131 A          SKG     AC0,XIMX         ; TEST FOR VALID INDEX (1 TO XIMX)
1F65 210C A          JMP     XFO2
1F66          ;
1F66          ;                  ... INVALID CROSS STREET COMMAND INDEX ...
1F66 8881 A XCCE:    LD      AC2,ATBUF
1F67 8200 A          LD      AC0,D0(AC2)      ; FETCH COMMAND INDEX WORD
1F68 886C A          LD      AC2,AXERR
1F69 A204 A          ST      AC0,D4(AC2)      ; STORE IN ERROR BUFFER, XERR1
1F6A 8203 A          LD      AC0,D3(AC2)      ; FETCH ERROR COUNTER AND INCREMENT
1F6B 4801 A          AISZ    AC0,C1
1F6C 2100 A          JMP     .+1
1F6D A203 A          ST      AC0,D3(AC2)
1F6E 8200 A          LD      AC0,D0(AC2)
1F6F 0721 A          SETBIT  D1               ; SET ERROR FLAG BIT 1
1F70 A200 A          ST      AC0,D0(AC2)
1F71 0200 A          RTS     D0
1F72          ;
1F72 8D11 A XFO2:    LD      AC3,AXCFT        ; FETCH ADDRESS OF COMMAND FORMAT TABLE
1F73 3300 A          RADD    AC0,AC3          ; ADD COMMAND INDEX
1F74 8300 A          LD      AC0,(AC3)        ; FETCH COMMAND FORMAT
1F75 072F A          SETBIT  D15              ; SET UPDATE BIT
1F76 8C6B A          LD      AC3,AXBUF
1F77 A302 A          ST      AC0,D2(AC3)      ; STORE IN XOCW2
1F78          ;
1F78 8881 A          LD      AC2,ATBUF
1F79 8C6B A          LD      AC3,AXBUF
1F7A 8303 A          LD      AC0,D3(AC3)      ; FETCH CURRENT AND STORE IN PREVIOUS
1F7B A304 A          ST      AC0,D4(AC3)      ;    COMMAND INDEX WORD
1F7C 8200 A          LD      AC0,D0(AC2)      ; FETCH AND SAVE CURRENT COMMAND
1F7D A303 A          ST      AC0,D3(AC3)      ;    INDEX WORD
1F7E 4C00 A          LI      AC0,C0
1F7F A200 A          ST      AC0,D0(AC2)          ; ZERO COMMAND INDEX WORD(TCIWC)
1F80 0200 A          RTS     D0
1F81          ;
1F81          ;                  ... LOCAL DATA ...
1F81 0006 A H0006:   .WORD   X'0006           ; HEX CONSTANT (BITS 1,2 SET)
1F82 007F A H007F:   .WORD   X'007F           ; HEX CONSTANT (BITS 15 - 8 RESET)
1F83 7F00 A H7F00:   .WORD   X'7F00           ; HEX CONSTANT (BITS 15 AND 7 - 0 RESET)
1F84          ;
```

```
1F84                    .LIST   LBX
1F84            ;
1F84            ;                       ... COMMAND FORMAT TABLE ...
1F84 1F84 A AXCFT:  .WORD   XCFT-1
1F85            XCFT:
1F85            ;                                           L
1F85            ;                                           T           S
1F85            ;                                                       E
1F85            ;                                           P   S       C   P
1F85            ;                                       U   R   P       O   R
1F85            ;                                       P   I   E       N   I
1F85            ;                                       D   M   C       D   M
1F85            ;                                       A   A   I   P   A   A
1F85            ;                                       T   R   A   E   R   R
1F85            ;                                       E   Y   L   D   Y   Y
1F85            ;
1F85            ;                                               C   C   C   C
1F85            ;                                               LS  LS  LS  LS
1F85            ;                                               ET  ET  ET  ET
1F85            ;                               COMMAND     GG GAO GAO GAO GAO
1F85            ;                                INDEX      OO ORP ORP ORP ORP
1F85            ;
1F85            ;                                       BITS SET = 1 (CIRCUITS ON)
1F85            ;                                       1 111 11
1F85            ;                                       5 432 109 876 543 210
1F85            ;
1F85            ;       ... PBD33J ...
1F85 0249 A             .WORD   X'0249          ; 1             S   S   S   S
1F86 0324 A             .WORD   X'0324          ; 2             S G   G   G       G GO
1F87 02A4 A             .WORD   X'02A4          ; 3             S  C  G   G       C CLEAR
1F88 0264 A             .WORD   X'0264          ; 4             S   S G   G       S STOP
1F89 0252 A             .WORD   X'0252          ; 5             S   S  C   C
1F8A            ;
1F8A            ;       ... PBD14C ...
1F8A 1124 A             .WORD   X'1124          ; 6         G     G   G   G
1F8B 10A4 A             .WORD   X'10A4          ; 7         G      C  G   G
1F8C 1094 A             .WORD   X'1094          ; 8         G      C   C  G
1F8D 304C A             .WORD   X'304C          ; 9        GG       S   S G
1F8E 044A A             .WORD   X'044A          ; A           C    S   S  C
1F8F 0921 A             .WORD   X'0921          ; B          G   G   G     S
1F90 1121 A             .WORD   X'1121          ; C         G    G   G     S
1F91 0249 A             .WORD   X'0249          ; D             S   S   S   S
1F92 0041 A             .WORD   X'0041          ; F                 S       S
1F93 0104 A             .WORD   X'0104          ; F                G       G
1F94 0084 A             .WORD   X'0084          ; 10                C       G
1F95 0082 A             .WORD   X'0082          ; 11                C        C
1F96            ;
1F96            XCFTE:                          ; END OF COMMAND FORMAT TABLE
1F96 0011 A XIMX:   .WORD   XCFTE-XCFT          ; MAXIMUM VALUE OF COMMAND INDEX
1F97            ;
```

```
1F97                .LIST   LBX
1F97        ;
1F97        ;               ... TURN STATUS LIGHTS ON/OFF ...
1F97        ;                   UPON ENTRY, AC3 CONTAINS PERIPHERAL
1F97        ;                                   DEVICE ADDRESS
1F97        ;
1F97        ;               ... TURN STATUS LIGHT 1 ON ...
1F97        XLON1:
1F97 4C00 A         LI      AC0,C0
1F98 0644 A         ROUT    FC44            ; OUTPUT CONTROL PULSE
1F99 0200 A         RTS     D0              ; RETURN FROM XLON1
1F9A        ;
1F9A        ;               ... TURN STATUS LIGHT 1 OFF ...
1F9A        XLOF1:
1F9A 4C00 A         LI      AC0,C0
1F9B 0645 A         ROUT    FC45            ; OUTPUT CONTROL PULSE
1F9C 0200 A         RTS     D0              ; RETURN FROM XLOF1
1F9D        ;
1F9D        ;               ... TURN STATUS LIGHT 2 ON ...
1F9D        XLON2:
1F9D 4C00 A         LI      AC0,C0
1F9E 0646 A         ROUT    FC46            ; OUTPUT CONTROL PULSE
1F9F 0200 A         RTS     D0              ; RETURN FROM XLON2
1FA0        ;
1FA0        ;               ... TURN STATUS LIGHT 2 OFF ...
1FA0        XLOF2:
1FA0 4C00 A         LI      AC0,C0
1FA1 0647 A         ROUT    FC47            ; OUTPUT CONTROL PULSE
1FA2 0200 A         RTS     D0              ; RETURN FROM XLOF2
```

DATA OUTPUT CPC - TURN STATUS LIGHTS ON/OFF

```
4401 1FA3                .PAGE
4402 1FA3                .ASECT
4403 1FA3 3000 A         .=X'3000
4404 3000        ;
```

```
3000                    .LIST   LDB
3000            ;
3000            ;                   ... D A T A   B A S E ...
3000            ;
3000            ;                   ... BUS DETECTOR SIGNATURE DATA ...
3000            ;
3000            ;         ( UP TO 32 DETECTORS @ 8 WORDS PER SIGNATURE)
3000            ;
3000            ;                       WORD 0 - TOTAL SIGNATURE DURATION
3000            ;                            1 - AMPLITUDE OF FIRST PEAK
3000            ;                            2 - TIME OF FIRST PEAK
3000            ;                            3 - AMPLITUDE OF SECOND PEAK
3000            ;                            4 - TIME OF SECOND PEAK
3000            ;                            5 - AMPLITUDE OF THIRD OR LAST PEAK
3000            ;                            6 - TIME OF LAST PEAK
3000            ;                            7 - END OF SIGNATURE STATUS
3000            ;
3000            VCSDB:
3000 3100 A             .=.+256
3100            ;
3100            ;                   ... DATA INPUT CPC DATA BASE ...
3100            BBUF:
3100 3101 A BDIN:       .=.+1                   ; DISCRETE INPUT DATA WORD
3101 3102 A BCR1:       .=.+1                   ; COMMAND REGISTER 1 STATUS
3102 3103 A BCR2:       .=.+1                   ; COMMAND REGISTER 2 STATUS
3103            ;
3103            ;                   ... PROG PANEL INPUT BUFFER ...
3103            ;
3103            ;                   ... DISCRETE SWITCHES ...
3103            ;
3103            ;                           DIAL 1
3103            ;                       BIT 0, BUS PREEMPT
3103            ;                       BIT 1, MASTER CONTROLLER SYNCH
3103            ;
3103            ;                           DIAL 2
3103            ;                       BIT 4, BUS PREEMPT
3103            ;                       BIT 5, MASTER CONTROLLER SYNCH
3103            ;
3103            ;                           DIAL 3
3103            ;                       BIT 8, BUS PREEMPT
3103            ;                       BIT 9, MASTER CONTROLLER SYNCH
3103            ;
3103            ;                       BIT 10, INPUT REQUEST
3103            ;                       BIT 11, MANUAL OVERRIDE
3103 3104 A BPPD:       .=.+1
3104            ;
3104            ;                   ... THUMBWHEEL INPUT DATA ...
3104            ;
3104            ;                       D0, CYCLE LENGTH
3104            ;                       D1, OFFSET 1
3104            ;                       D2, OFFSET 2
```

```
3104          ;                                   D3, OFFSET 3
3104          ;                                   D4, MSG SPLIT (%)
3104          ;                                   D5, INT 1
3104          ;                                   D6, INT 2
3104 310B A BPD1:   .=.+7                             ; DIAL 1
310B 3112 A BPD2:   .=.+7                             ; DIAL 2
3112 3119 A BPD3:   .=.+7                             ; DIAL 3
3119          ;
3119          ;                          ... DATA OUTPUT CPC DATA BASE ...
3119          XBUF:
3119 311A A XSDW:   .=.+1                             ; STATUS DISPLAY WORD
311A 311B A XOCW1:  .=.+1                             ; OUTPUT COMMAND WORD 1 (MAIN STREET)
311B 311C A XOCW2:  .=.+1                             ; OUTPUT COMMAND WORD 2 (CROSS STREET)
311C 311D A XCIWC:  .=.+1                             ; CURRENT COMMAND INDEX WORD
311D 311E A XCIWP:  .=.+1                             ; PREVIOUS COMMAND INDEX WORD
311E          ;
311E          ;                          ... PREEMPT CPC DATA BASE ...
311E          PBUF:
311E 311F A PIP:    .=.+1                                 ; PREEMPT IN PROGRESS FLAG
311F 3120 A PTEM:   .=.+1                                 ; TEMPORARY STORAGE LOCATION
3120 3121 A PTCP:   .=.+1                             ; PRIMARY CONTRACTIBLE INT ADDRESS
3121 3122 A PTCS:   .=.+1                             ; SECONDARY CONTRACTIBLE INT ADDRESS
3122 3123 A PTEI:   .=.+1                             ; EXTENDIBLE INT ADDRESS
3123 3126 A PMIF:   .=.+3                                 ; PREEMPT MONITOR INDEX FLAGS
3126 3127 A PFREQ:  .=.+1                                 ; PREEMPT REQUEST FLAG
3127 3128 A PMDT:   .=.+1                                 ; PREEMPT MINIMUM DURATION TIMER
3128 3129 A PTAVT:  .=.+1                                 ; AVAILIBLE TIME FOR INTERVAL RESTORATION
3129 312A A PTITE:  .=.+1                                 ; INTERVAL TO BE RESTORED
312A 312B A PTPMT:  .=.+1                                 ; FIRST LOCATION OF MASTER TIMING PLAN
312B          ;
312B          ;                          ... MAXIMUM PREEMPT EXTENSION PARAMETERS ...
312B 312C A PEMAX:  .=.+1                             ; CURRENT DIAL
312C 312D A         .=.+1                             ; DIAL 1
312D 312E A         .=.+1                             ; DIAL 2
312E 312F A         .=.+1                             ; DIAL 3
312F          ;
312F          ;                          ... VEHICLE CLASSIFIER DATA ...
312F 3130 A VCDT:   .=.+1                                 ; VEHICLE DETECTOR INDEX
3130 3131 A VCBT:   .=.+1                             ; TIMER FOR RECENT BUS CLASS.
3131          ;
3131 3151 A VCVC:   .=.+32                            ; VEHICLE COUNTERS (ONE PER DETECTOR)
3151 3165 A VCDB:   .=.+20
3165          ;
3165          ;                          ... VEHICLE CLASSIFIER VEHICLE LOG ...
3165          ;
3165          ;                                 DOUBLE PRECISION, UNSIGNED WORDS
3165          ;
3165          VCLOG:
3165 3167 A         .=.+2                                 ; NUMBER OF SIGNATURES REJECTED AS NOISE
3167 3169 A         .=.+2                                 ; NUMBER OF GROSS TEST FAILURES
3169 316B A         .=.+2                                 ; NUMBER OF VEHICLES HANDLED BY
316B          ;                                             STOPPED VEHICLE CLASSIFIER
```

```
316B 316D A          .=.+2                    ; NUMBER OF VEHICLES HANDLED BY
316D          ;                                 MOVING VEHICLE CLASSIFIER
316D 316F A          .=.+2                    ; NUMBER OF VEHICLES CLASSIFIED
316F          ;                                 AS BUSES
316F 3171 A          .=.+2                    ; NUMBER OF VEHICLES CLASSIFIED
3171          ;                                 AS NON-BUSES
3171          ;
3171          ;                    ... TIMING PLAN CPC DATA BASE ...
3171          TBUF:
3171 3172 A TCIWC:   .=.+1                    ; COMMAND INDEX WORD BUFFER
3172 3173 A TCDS:    .=.+1                    ; CURRENT DIAL SELECTION
3173 3174 A TIN:     .=.+1                    ; CURRENT INTERVAL INDEX
3174 3175 A TINT:    .=.+1                    ; CURRENT INTERVAL TIMER
3175 3176 A TFST:    .=.+1                    ; WAIT FOR SYNC START FLAG
3176 3177 A TECC:    .=.+1                    ; END OF CYCLE FLAG
3177 3178 A TEMP:    .=.+1                    ; TEMPORARY LOCATION
3178 3179 A TDWL:    .=.+1                    ; DIAL 1, MAX DWELL FOR SYNC START
3179 317A A          .=.+1                    ; DIAL 2, MAX DWELL FOR SYNC START
317A 317B A          .=.+1                    ; DIAL 3, MAX DWELL FOR SYNC START
317B 318F A TPM1:    .=.+20                   ; MASTER TP TABLE, DIAL 1
318F 31A3 A TPM2:    .=.+20                   ; MASTER TP TABLE, DIAL 2
31A3 31B7 A TPM3:    .=.+20                   ; MASTER TP TABLE, DIAL 3
31B7 31CB A TNTP:    .=.+20                   ; NEXT TP TABLE
31CB 31DF A TCTP:    .=.+20                   ; CURRENT TP TABLE
31DF          ;
31DF          ;                    ... EXECUTIVE CPC DATA BASE ...
31DF          ;
31DF          EBUF:
31DF 31E0 A ERTCF:   .=.+1                    ; RTC COMPUTE FLAG
31E0          ;
31E0 31E1 A ETST:    .=.+1                    ; 'WAIT FOR SYNCH START' DWELL TIMER
31E1          ;
31E1 31E2 A ETCY:    .=.+1                    ; CURRENT CYCLE TIMER
31E2 31E3 A          .=.+1                    ; PREVIOUS MAXIMUM
31E3 31E4 A          .=.+1                    ; PREVIOUS MINIMUM
31E4 31E5 A EPST:    .=.+1                    ; START PULSE INDICATOR
31E5 31E6 A EPDT:    .=.+1                    ; START PULSE DURATION TIMER
31E6          ;
31E6 31E7 A ECMF:    .=.+1                    ; TEMPORARY FLAG FOR ECOM
31E7          ESYD:                           ; SYNCH DATA BUFFER (DIAL 1,2,3)
31E7                                          ;    WORD 0 - CYCLE LENGTH
31E7                                          ;    WORD 1 - RESET 1 POSITION ON DIAL
31E7                                          ;    WORD 2 - RESET 2 POSITION ON DIAL
31E7                                          ;    WORD 3 - RESET 3 POSITION ON DIAL
31E7                                          ;    WORD 4 - END OF START PULSE (3 %)
31E7                                          ;    WORD 5 - SYNCH DIAL TIMER
31E7 31ED A          .=.+6                    ; DIAL 1
31ED 31F3 A          .=.+6                    ; DIAL 2
31F3 31F9 A          .=.+6                    ; DIAL 3
31F9          ;
31F9          ;                    ... LOOP MASTER INTERCONNECT DATA ...
31F9 31FA A ER123:   .=.+1                    ; RESET SELECT
```

```
31FA 31FB A ERN0:   .=.+1          ; ERN0, 0TH PAST VALUE OF R1,R2,R3
31FB 31FC A         .=.+1          ; ERN1, 1ST PAST VALUE OF R1,R2,R3
31FC 31FD A         .=.+1          ; ERN2, 2ND PAST VALUE OF R1,R2,R3
31FD        ;
31FD 31FE A ED23:   .=.+1          ; DIAL SELECT
31FE 31FF A EDN0:   .=.+1          ; EDN0, 0TH PAST VALUE OF D2,D3
31FF 3200 A         .=.+1          ; EDN1, 1ST PAST VALUE OF D2,D3
3200 3201 A         .=.+1          ; EDN2, 2ND PAST VALUE OF D2,D3
3201        ;
3201 3202 A EDMIT:  .=.+1          ; DETECTOR MONITORING INTERVAL TIMER
3202 3203 A EDETT:  .=.+1          ; TEMP STORAGE FOR DET MONITORING
3203        ;
```

```
3203                .LIST   LDB
3203        ;
3203        ;                   ... SAVE AREA FOR RTC INTERRUPTS ...
3203        ESAV:
3203 3204 A         .=.+1                   ; AC0
3204 3205 A         .=.+1                   ; AC1
3205 3206 A ESV2:   .=.+1                   ; AC2
3206 3207 A         .=.+1                   ; AC3
3207 3208 A ESVF:   .=.+1                   ; RALU FLAGS
3208 3218 A         .=.+16                  ; STACK SAVE AREA
3218 3219 A ESVS:   .=.+1                   ; SELECT FLAG STATE
3219        ;
3219        ;                   ... SNAPSHOT (COTF) SAVE AREA ...
3219        CSAV:
3219 321A A         .=.+1                           ; AC0
321A 321B A         .=.+1                           ; AC1
321B 321C A CSV2:   .=.+1                           ; AC2
321C 321D A         .=.+1                           ; AC3
321D 321E A CSVF:   .=.+1                           ; RALU FLAGS
321E 322E A         .=.+16                          ; STACK SAVE AREA
322E 322F A CSVS:   .=.+1                           ; SELECT FLAG STATE
322F        ;
322F        ;                   ... STACKFULL SAVE AREA ...
322F        ISV:
322F 3230 A ISV0:   .=.+1
3230 3231 A ISV1:   .=.+1
3231 3232 A ISV2:   .=.+1
3232 3233 A ISV3:   .=.+1
3233 3234 A ISTK1:  .=.+1
3234 3235 A ISTK2:  .=.+1
3235 3236 A ISTK3:  .=.+1
3236 3237 A ISTK4:  .=.+1
3237 3243 A ISTK5:  .=.+12
3243        ;
3243        ;                   ... DISCONNECT SAVE AREA ...
3243        EDSAV:
3243 3244 A         .=.+1                   ; AC0
3244 3245 A         .=.+1                   ; AC1
3245 3246 A EDSV2:  .=.+1                   ; AC2
3246 3247 A         .=.+1                   ; AC3
3247 3248 A EDSVF:  .=.+1                   ; RALU FLAGS
3248 3258 A         .=.+16                  ; STACK SAVE AREA
3258 3259 A EDSVS:  .=.+1                   ; SELECT FLAG STATE
3259        ;
3259        ;                   ... OPTIONAL USE SAVE AREA ...
3259 3261 A CTST:   .=.+8                           ; OPTIONAL USE SAVE AREA
```

```
3261                    .LIST   LDB
3261          ;
3261          ;                    ... PROG. PANEL DATA ...
3261          ;
3261                                  PANEL ROUTINE WORKING BUFFER
3261 3268 A EPD1:   .=.+7                  ; (SAME ALLOCATION AS BPD1)
3268 326F A EPD2:   .=.+7                  ; (SAME ALLOCATION AS BPD2)
326F 3276 A EPD3:   .=.+7                  ; (SAME ALLOCATION AS BPD3)
3276          ;
3276          ;                             PANEL CONVERTED DATA UPDATE BUFFER
3276 3277 A EPCD1:  .=.+1                  ; FIRST ADJ. INT. LENGTH, DIAL 1
3277 3278 A         .=.+1                  ; SECOND ADJ. INT. LENGTH, DIAL 1
3278 3279 A         .=.+1                  ; DELTA EMAX,              DIAL 1
3279 327A A         .=.+1                  ; CYCLE LENGTH             DIAL 1
327A 327B A         .=.+1                  ; RESET 1                  DIAL 1
327B 327C A         .=.+1                  ; RESET 2                  DIAL 1
327C 327D A         .=.+1                  ; RESET 3                  DIAL 1
327D 327E A         .=.+1                  ; START PULSE DURATION     DIAL 1
327E 327F A         .=.+1                  ; MAX DWELL AT MSG         DIAL 1
327F 3280 A         .=.+1                  ; INT2                     DIAL 1
3280 328A A EPCD2:  .=.+10                 ;                          DIAL 2
328A 3294 A EPCD3:  .=.+10                 ;                          DIAL 3
3294          ;
3294          ;                             FIXED TIME IN CYCLE
3294 3295 A EPFT:   .=.+1                  ; SUM OF FIXED INTERVALS   DIAL 1
3295 3296 A         .=.+1                  ; SUM OF FIXED INTERVALS   DIAL 2
3296 3297 A         .=.+1                  ; SUM OF FIXED INTERVALS   DIAL 3
3297          ;
3297 3298 A EP1E:   .=.+1                 ; ERROR FLAG FOR PLAN 1
3298 3299 A EP2E:   .=.+1                 ; ERROR FLAG FOR PLAN 2
3299 329A A EP3E:   .=.+1                 ; ERROR FLAG FOR PLAN 3
329A 329B A EPDVE:  .=.+1                 ; ERROR FLAG FOR DATA VALIDITY
329B          ;
329B 329C A EPUF:   .=.+1                 ; PROG PANEL UPDATE FLAG
329C 329D A EPCF:   .=.+1                 ; PROG PANEL COMPUTATION FLAG
329D 329E A EPIR:   .=.+1                 ; STATE OF PROG PANEL INPUT REQ SWITCH
```

```
329E                    .LIST   LDB
329E            ;
329E            ;                       ... SELF TEST DATA BASE ...
329E 329F A SSAD:       .=.+1                   ; RAM ADDRESS TEST START ADDRESS
329F 32A0 A SEAD:       .=.+1                   ; RAM ADDRESS TEST END ADDRESS
32A0 32BE A             .=.+30                  ; (EXPANSIBILITY)
32BE            ;
```

```
32BE                .LIST   LDB
32BE        ;
32BE        ;                   ... E R R O R   B U F F E R ...
32BE        ;
32BE        ;                   ... DATA INPUT CPC ERROR FLAGS, COUNTERS, DATA ...
32BE        BERR:
32BE 32BF A BERRF:  .=.+1                   ; ERROR FLAGS
32BF        ;
32BF        ;                   ... DATA OUTPUT CPC ERROR FLAGS, COUNTERS, DATA ...
32BF        XERR:
32BF 32C0 A XERRF:  .=.+1                   ; ERROR FLAGS (SET CONDITION, BIT = 1)
32C0        ;                                   BIT 0, INVALID COMMAND INDEX FOR
32C0        ;                                          MAIN STREET
32C0        ;                                   BIT 1, INVALID COMMAND INDEX FOR
32C0        ;                                          CROSS STREET
32C0 32C2 A XERR0:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32C2 32C4 A XERR1:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32C4        ;
32C4        ;                   ... PREEMPT CPC ERROR FLAGS, COUNTERS, DATA ...
32C4        PERR:
32C4 32C5 A PERRF:  .=.+1                   ; ERROR BUFFER FLAG (SET CONDITION, BIT = 1)
32C5        ;                                   BIT 0, ARITHMETIC OVERFLOW
32C5        ;                                   BIT 1, OUT OF RANGE (VCLASS DETECTOR INDEX)
32C5        ;                                   BIT 2, DATA EXCEEDS BOUNDS (VEHICLE AMP)
32C5        ;                                   BIT 3, DATA EXCEEDS BOUNDS (VEHICLE TIME)
32C5        ;                                   BIT 4, DATA SEQUENCE ERROR (VEHICLE TIME)
32C5        ;                                   BIT 5, DATA VALIDITY ERROR (NOISE)
32C5        ;                                   BIT 6, EOS STATUS WORD ALTERED DURING COPY
32C5        ;                                          OPERATION
32C5        ;                                   BIT 7, SIGNATURE LOST DURING COPY OPERATION
32C5        ;                                   BIT 8, OUT OF RANGE (PREQ DETECTOR INDEX)
32C5        ;                               ... PBD33J ...
32C5        ;                                   BIT 9, DATA VALIDITY ERROR ( ATTEMPT
32C5        ;                                          TO PREEMPT DOWNSTREAM VEHICLE)
32C5        ;                               ... PBD14C ...
32C5        ;                                   BIT 9, NON-POSITIVE DURATION OF SOURCE
32C5        ;                                          EXCHANGE INTERVAL AT PREEMPT MOD
32C5        ;
32C5 32D9 A         .=.+20                      ; (ERROR COUNTER, ERROR WORD)
32D9        ;
32D9        ;                   ... TIMING PLAN CPC ERROR FLAGS, COUNTERS, DATA ...
32D9        TERR:
32D9 32DA A TERRF:  .=.+1                   ; ERROR FLAGS (SET CONDITION, BIT = 1)
32DA        ;                                   BIT 0, DIAL SELECT ERROR
32DA        ;                                   BIT 1, INTERVAL INDEX OUT OF RANGE
32DA 32DC A TERR0:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32DC 32DE A TERR1:  .=.+2                   ; ERROR COUNTER, ERROR WORD
32DE        ;
32DE        ;                   ... EXECUTIVE CPC ERROR FLAGS, COUNTERS, DATA ...
32DE        EERR:
32DE 32DF A EERRF:  .=.+1                   ; ERROR FLAGS (SET CONDITION, BIT = 1)
32DF        ;                                   BIT 0, RTC ERROR
```

```
32DF            ;                                   BIT 1, DCR1.NE.DOCW1
32DF            ;                                   BIT 2, DCR2.NE.DOCW2
32DF            ;                                   BIT 3, MORE THAN ONE RESET HIGH
32DF            ;                                   BIT 4, DETECTOR UNDERCOUNTING
32DF                                        ;             (TOTAL FOR ALL DETECTORS,
32DF                                        ;              INDEX OF LAST DETECTOR)
32DF            ;                                   BIT 5, INVALID DET INDEX
32DF 32E1 A EERR0:   .=.+2                  ; ERROR COUNTER, ERROR WORD
32E1 32E3 A EERR1:   .=.+2                  ; ERROR COUNTER, ERROR WORD
32E3 32E5 A EERR2:   .=.+2                  ; ERROR COUNTER, ERROR WORD
32E5 32E7 A EERR3:   .=.+2                  ; ERROR COUNTER, ERROR WORD
32E7 32E9 A EERR4:   .=.+2                  ; ERROR COUNTER, ERROR WORD
32E9 32EB A EERR5:   .=.+2                  ; ERROR COUNTER, ERROR WORD
32EB            ;
32EB            ;                     ... PROG. PANEL ERROR DATA ...
32EB            ;
32EB            ;                           ERROR FLAGS (SET CONDITION, BIT = 1)
32EB            ;                              BIT 0, ERROR IN ADJ INT OF TP MASTER
32EB            ;                              BIT 1, DATA VALIDITY ERROR (CYCLE LENGTH)
32EB            ;                              BIT 2, DATA VALIDITY ERROR (OFFSET 1)
32EB            ;                              BIT 3, DATA VALIDITY ERROR (OFFSET 2)
32EB            ;                              BIT 4, DATA VALIDITY ERROR (OFFSET 3)
32EB            ;                              BIT 5, DATA VALIDITY ERROR (MSG SPLIT %)
32EB            ;                              BIT 6, DATA VALIDITY ERROR (INT1)
32EB            ;                              BIT 7, DATA VALIDITY ERROR (INT2)
32EB            ;                              BIT 8,  OUT OF RANGE (CYCLE LENGTH)
32EB            ;                              BIT 9,  OUT OF RANGE (OFFSET 1)
32EB            ;                              BIT 10, OUT OF RANGE (OFFSET 2)
32EB            ;                              BIT 11, OUT OF RANGE (OFFSET 3)
32EB            ;                              BIT 12, OUT OF RANGE (INT1)
32EB            ;
32EB            ;
32EB 32EC A EPEF1:   .=.+1                  ; ERROR FLAGS
32EC 3306 A          .=.+26                 ; (ERROR COUNTER, ERROR WORD)
3306            ;
3306 3307 A EPEF2:   .=.+1                  ; ERROR FLAGS
3307 3321 A          .=.+26                 ; (ERROR COUNTER, ERROR WORD)
3321            ;
3321 3322 A EPEF3:   .=.+1                  ; ERROR FLAGS
3322 333C A          .=.+26                 ; (ERROR COUNTER, ERROR WORD)
333C            ;
333C            ;                     ... DETECTOR ERROR COUNTERS, DATA ...
333C            EDERR:
333C 337C A          .=.+64                 ; (ERROR COUNTER, ERROR DATUM)
337C                                        ;    FOR 32 DETECTORS
337C            ;
337C            ;
337C 337D A DEND:    .=.+1                  ; LAST LOCATION IN READ/WRITE DATA BASE
```

```
4772 337D                .PAGE   '   '
4773 337D                .LIST   1
4774 337D 0001 A LAC     =       1
4775 337D 0001 A LBP     =       1
4776 337D 0001 A LER     =       1
4777 337D 0001 A LES     =       1
4778 337D 0001 A LPP     =       1
4779 337D 0001 A LTP     =       1
4780 337D 0001 A LPR     =       1
4781 337D 0001 A LVC     =       1
4782 337D 0001 A LEB     =       1
4783 337D 0001 A LBX     =       1
4784 337D 0001 A LDB     =       1
4785 337D 0001 A LDU     =       1
4786 337D                .EXTD
4787 337D                .END

  ******      0 ERRORS IN ASSEMBLY   ******
```

```
ABPUF  ABDIN  ABERR  ABLSMU ABPAN  ABPD1  ABPD2  ABPD3  ABPPD  ABRTC
0068 A 0069 A 006A A 00E1 A 003E A 00BC A 00BD A 00BE A 00BB A 003C A

AC0    AC1    AC2    AC3    ACCU   ACCUI  ACDISP ACMU   ACOTF  ACRAM
0000 A 0001 A 0002 A 0003 A 00E8 A 10DA A 10DC A 00E2 A 003D A 10DD A

ACSAV  ACSV2  ACSVF  ACSVS  ACTST  ACWAP  ADEND  ADMU   AE10   AE100
00AE A 00AF A 00B0 A 00B1 A 00BA A 10DE A 00D6 A 00E5 A 00CC A 00CB A

AEADC  AEBKD  AEBUF  AECCU  AECCUR AECER  AECMF  AECOM  AED23  AEDERR
002F A 0024 A 009A A 0022 A 10DF A 0025 A 00A0 A 0028 A 00A4 A 00A8 A

AEDET  AEDETT AEDIS  AEDMIT AEDNO  AEDSV2 AEDSVF AEDSVS AEFOC  AEERR
002E A 00A7 A 0027 A 00A6 A 00A5 A 00B7 A 00B8 A 00B9 A 002C A 009F A

AEGIS  AEM10  AEM24  AEMAD  AEMIC  AEP1E  AEP2E  AEP3E  AEPAN  AEPBI2
0021 A 0047 A 0048 A 0023 A 002A A 00CD A 00CE A 00CF A 003F A 0045 A

AEPBI3 AEPCD1 AEPCD2 AEPCD3 AEPCF  AEPCY  AEPD1  AEPD2  AEPD3  AEPDT
0046 A 00C2 A 00C3 A 00C4 A 00C7 A 00C9 A 00BF A 00C0 A 00C1 A 009E A

AEPDVE AEPEF1 AEPEF2 AEPEF3 AEPFT  AEPIN1 AEPIR  AEPP1  AEPP2  AEPP3
00D0 A 00D1 A 00D2 A 00D3 A 00C5 A 00CA A 00C8 A 0041 A 0042 A 0043 A

AEPPM  AEPST  AEPUF  AEPUP  AER123 AERAM  AERE0  AERE1  AERE10 AERE11
0040 A 009D A 00C6 A 0044 A 00A2 A 00A9 A 0049 A 004A A 0053 A 0054 A

AERE12 AERE2  AERE3  AERE4  AERE5  AERE6  AERE7  AERE8  AERE9  AERNO
0055 A 004B A 004C A 004D A 004E A 004F A 0050 A 0051 A 0052 A 00A3 A

AERSI  AESAV  AESCH  AESDW  AESV2  AESVF  AESVS  AESYD  AESYN  AETCY
0020 A 00AA A 0026 A 002D A 00AB A 00AC A 00AD A 00A1 A 0029 A 009B A

AETIM  AETST  AISTK1 AISTK2 AISTK3 AISTK4 AISV2  ALDU   APBUF  APCIM
002B A 009C A 00B3 A 00B4 A 00B5 A 00B6 A 00B2 A 00E0 A 006D A 0034 A

APEMAX APERR  APFREQ APIP   APMDT  APMIF  APMON  APP1U  APP2U  APP3U
0076 A 0077 A 0074 A 006E A 0075 A 0073 A 0030 A 00EA A 00EB A 00EC A

APPDU  APREQ  APSWCH APTAVT APTCP  APTCS  APTEI  APTEM  APTFR  APTITE
00E9 A 0032 A 1A4F A 0078 A 0070 A 0071 A 0072 A 006F A 0033 A 0079 A

APTPMT ARTC   ASDB   ASDU   ASEAD  ASMAD  ASSAD  ATAI1  ATAI2  ATBUF
007A A 00E7 A 00E6 A 00E4 A 00D5 A 005A A 00D4 A 0093 A 0094 A 0081 A

ATCDS  ATCIW  ATCIWC ATCOPY ATCP   ATCTP  ATDIAL ATDWL  ATFCI  ATEMP
0083 A 0097 A 0082 A 0036 A 00E3 A 008F A 0038 A 0089 A 0096 A 0088 A

ATEOC  ATERR  ATFST  ATHI   ATIMX  ATIN   ATINT  ATNCY  ATNTP  ATPE
0087 A 0090 A 0086 A 0092 A 0095 A 0084 A 0085 A 0037 A 008F A 0099 A
```

```
ATPEX  ATPM1  ATPM2  ATPM3  ATPP   ATPTAB ATSCI  AVCBT  AVCBUS AVCDB
0035 A 008B A 009C A 008D A 008A A 0098 A 0091 A 007B A 1D23 A 007F A

AVCDT  AVCER  AVCK   AVCLAS AVCLOG AVCSDB AVCVC  AVCVNB AXBUF  AXCFT
007E A 1D25 A 1D91 A 0031 A 007D A 0080 A 007C A 1D24 A 006B A 1F84 A

AXERR  AXLOF1 AXLOF2 AXLON1 AXLON2 AXRTC  AXSDC  AXWDT  BBUF   BCR1
006C A 0057 A 0059 A 0056 A 0058 A 0039 A 003B A 003A A 3100 A 3101 A

BCR2   BDIN   BERR   BERRF  BPAN   BPD1   BPD2   BPD3   BPIN   BPPD
3102 A 3100 A 325E A 32BE A 1F0D A 3104 A 310B A 3112 A 1F19 A 3103 A

BRTC   C0     C1     C125   C15    C2     C3     C4     C48    C5
1F00 A 0000 A 0001 A 007D A 000F A 0002 A 0003 A 0004 A 0030 A 0005 A

C6     C7     C8     CCB0   CCB1   CCCF   CCCU   CCEZ   CCNZ   CCPH
0006 A 0007 A 0008 A 0003 A 0004 A 000E A 000C A 0001 A 0005 A 000D A

CCSF   CCZN   CCZP   CM1    CM10   CM100  CM110  CM16   CM2    CM20
0008 A 000B A 0002 A FFFF A FFF6 A FF9C A FF92 A FFF0 A FFFE A FFEC A

CM6    COTF   COTF1  COTF2  COTF3  COTF4  COTF5  CSAV   CSV2   CSVF
FFFA A 1143 A 114E A 115F A 1161 A 1172 A 1175 A 3219 A 321B A 321D A

CSVS   CTST   D0     D1     D10    D11    D116   D118   D12    D13
322E A 3259 A 0000 A 0001 A 000A A 000B A 0074 A 0076 A 000C A 000D A

D14    D15    D16    D17    D18    D19    D2     D20    D21    D22
000E A 000F A 0010 A 0011 A 0012 A 0013 A 0002 A 0014 A 0015 A 0016 A

D23    D24    D25    D26    D3     D4     D5     D6     D7     D8
0017 A 0018 A 0019 A 001A A 0003 A 0004 A 0005 A 0006 A 0007 A 0008 A

D9     DEND   DETS   DM1    DM12   DM2    DM4    DM7    DPMD   E10
0009 A 337C A 0001 A FFFF A FFF4 A FFFE A FFFC A FFF9 A 0019 A 146C A

E100   EADC   EADC1  EADC2  EADC3  EADC4  EBKD   EBKD1  EBKDE  EBUF
146D A 110F A 111F A 1130 A 1132 A 113B A 1E00 A 1E00 A 1E0A A 31DF A

ECCU   ECCU1  ECCUI  ECCUR  ECER   ECMF   ECOM   ECOM1  ECOM2  ECOMR
10C2 A 10CB A 10DB A 10D9 A 10BF A 31E6 A 1212 A 1227 A 123B A 123E A

ED23   EDERR  EDET   EDET1  EDET2  EDET3  EDET32 EDETI  EDETM  EDETR
31FD A 333C A 12B2 A 12B6 A 12C6 A 12D3 A 12DE A 12DC A 12DD A 12DB A

EDETT  EDIS   EDISH  EDLAY  EDMIT  EDNO   EDSAV  EDSV2  EDSVF  EDSVS
3202 A 1200 A 120F A 110F A 3201 A 31FE A 3243 A 3245 A 3247 A 3258 A

EEOC   EERR   EERR0  EERR1  EERR2  EERR3  EERR4  EERR5  EERRF  EGIS
131B A 32DE A 32DF A 32E1 A 32E3 A 32E5 A 32E7 A 32E9 A 32DF A 1041 A
```

EGIS1	EGIS2	EGIS3	EGIS4	EGIS5	EHLT	EM10	EM24	EMAD	EMADE
104D A	105E A	108B A	10A7 A	10AA A	1039 A	174D A	1752 A	10E0 A	10EF A
EMADR	EMIC	EMIC1	EMIC2	EP1E	EP2E	EP3E	EPAN	EPAN1	EPANE
10F2 A	1282 A	1291 A	12A0 A	3297 A	3298 A	3299 A	1400 A	1426 A	1420 A
EPANE1	EPBI	EPBI1	EPBI2	EPBI3	EPBIR	EPCD1	EPCD2	EPCD3	EPCF
1422 A	1726 A	173E A	1733 A	1726 A	1745 A	3276 A	3280 A	328A A	329C A
EPCY	EPD1	EPD2	EPD3	EPDT	EPDVE	EPEF1	EPEF2	EPEF3	EPEOC
146A A	3261 A	3268 A	326F A	31E5 A	329A A	32EB A	3306 A	3321 A	1321 A
EPFI	EPFI1	EPFIR	EPFT	EPIN1	EPIR	EPP1	EPP10	EPP11	EPP12
1461 A	1462 A	1469 A	3294 A	146B A	329D A	14C4 A	14F3 A	150C A	1525 A
EPP13	EPP14	EPP15	EPP2	EPP20	EPP21	EPP22	EPP23	EPP24	EPP25
153E A	155B A	1572 A	1578 A	15A7 A	15C0 A	15D9 A	15F2 A	160F A	1626 A
EPP3	EPP30	EPP31	EPP32	EPP33	EPP34	EPP35	EPPM	EPPM1	EPPM2
162C A	165B A	1674 A	168D A	16A6 A	16C3 A	16DA A	146E A	1478 A	148B A
EPPM3	EPPM4	EPPM5	EPPMR	EPST	EPUF	EPUP	EPUPS	ER123	ERE0
1499 A	14A7 A	14B5 A	14C2 A	31E4 A	329B A	16E0 A	171A A	31F9 A	1340 A
ERE1	ERE10	ERE11	ERE12	ERE2	ERE3	ERE4	ERE5	ERE6	ERE7
1349 A	139A A	13A3 A	13AC A	1352 A	135B A	1364 A	136D A	1376 A	137F A
ERE8	ERE9	ERNO	ERSI	ERTC	ERTC1	ERTCF	ERTS	ESAV	ESCH
1383 A	1391 A	31FA A	1000 A	108B A	1098 A	31DF A	110D A	3203 A	10F3 A
ESDW	ESDW1	ESTK	ESTK1	ESV2	ESVF	ESVS	ESYA	ESYA1	ESYA2
12E0 A	1301 A	1061 A	1078 A	3205 A	3207 A	3218 A	1279 A	127C A	1281 A
ESYD	ESYN	ESYN1	ESYN2	ESYN3	ESYN4	ESYN4A	ESYN5	ESYN6	ETCPY
31E7 A	123F A	1257 A	125C A	1261 A	1267 A	1266 A	1272 A	1278 A	1023 A
ETCY	ETEOC	ETEOC1	ETEOC2	ETIM	ETIM1	ETIMR	ETST	EVCVC	EZDB
31E1 A	1333 A	1339 A	133D A	12A3 A	12AD A	12B1 A	31E0 A	12DF A	101C A
FC1	FC10	FC11	FC12	FC13	FC14	FC15	FC17	FC2	FC3
0001 A	0010 A	0011 A	0012 A	0013 A	0014 A	0015 A	0017 A	0002 A	0003 A
FC4	FC41	FC42	FC43	FC44	FC45	FC46	FC47	H0006	H0009
0004 A	0041 A	0042 A	0043 A	0044 A	0045 A	0046 A	0047 A	1F81 A	174C A
H000E	H000F	H0010	H0030	H007F	H0090	H00F0	H0200	H0400	H0900
12A1 A	1749 A	131A A	12A2 A	1F82 A	174B A	1748 A	00F5 A	14C3 A	174A A
H0F00	H0FFF	H1T1	H1T2	H2T1	H2T2	H3000	H3BFF	H3FFF	H7F00
1747 A	1040 A	1C8E A	1C8F A	1C90 A	1C91 A	00F1 A	00F6 A	00F2 A	1F83 A

H7FFF	HFC00	HECCF	HF800	HFFEF	INEN	ISTK1	ISTK2	ISTK3	ISTK4
00F0 A	00F3 A	00F4 A	1318 A	1319 A	0001 A	3233 A	3234 A	3235 A	3236 A
ISTK5	ISV	ISV0	ISV1	ISV2	ISV3	LAC	LBP	LBX	LDB
3237 A	322F A	322F A	3230 A	3231 A	3232 A	0001 A	0001 A	0001 A	0001 A
LDU	LEB	LER	LES	LPP	LPR	LTP	LVC	PBUF	PCIM
0001 A	0001 A	0001 A	0001 A	0001 A	0001 A	0001 A	0001 A	311F A	1A4E A
PEMAX	PERR	PERRF	PFREQ	PIP	PM0	PM01	PM02	PM05	PM1
3128 A	32C4 A	32C4 A	3126 A	311E A	1A0F A	1A10 A	1A1A A	1A2B A	1A50 A
PM2	PM3	PM4	PM5	PMAS	PMAS2	PMDT	PMER10	PMHD	PMIF
1A53 A	1A8B A	1A8C A	1A8E A	1A48 A	1A4C A	3127 A	1A44 A	1A4F A	3123 A
PMM1	PMON	PMRET	PREQ	PSWCH	PSWFC	PSWFD	PTAVT	PTCP	PTCS
1A4D A	1A00 A	1A47 A	1AE6 A	1A98 A	1AAF A	1AB0 A	3128 A	3120 A	3121 A
PTEI	PTEM	PTER	PTITE	PTPMT	SEAD	SELF	SMAD	SMAD1	SMAD2
3122 A	311F A	1A88 A	3129 A	312A A	329F A	0002 A	1E0C A	1E0F A	1E14 A
SMAD3	SMADE	SSAD	T0	T1	T2	T3	TAI1	TAI2	TBUF
1E1C A	1E19 A	329E A	182D A	182E A	182F A	1830 A	18D5 A	18D6 A	3171 A
TCDS	TCIW	TCIWC	TCOPY	TCTP	TD1	TD2	TD3	TDCH	TDIAL
3172 A	1918 A	3171 A	18C9 A	31CB A	18B0 A	18AA A	18AE A	18B4 A	1896 A
TDSEL	TDWL	TECI	TEMP	TEOC	TERR	TERR0	TERR1	TERRF	TEZ
18B1 A	3178 A	18D8 A	3177 A	3176 A	32D9 A	32DA A	32DC A	32D9 A	184B A
TFEP	TFST	TGEZ	TGZ	THI	TIMX	TIN	TINT	TLFP	TLOOP
18E0 A	3175 A	1847 A	1852 A	18D4 A	18D7 A	3173 A	3174 A	18F1 A	18CA A
TLZ	TMSG	TNCY	TND2	TNDC	TNTP	TNXT	TOL	TPF	TPEX
1842 A	180E A	1866 A	18AC A	18C0 A	31B7 A	1831 A	1859 A	1967 A	1800 A
TPM1	TPM2	TPM3	TPTAB	TPUE	TPSET	TSCI	TSD2	TSD3	TSST
317B A	318F A	31A3 A	192C A	1896 A	1829 A	18D3 A	1878 A	187F A	1818 A
TSYNC	TSYNCE	TSYNCR	VC01	VC02	VC31	VC32	VC41	VC42	VCBT
1870 A	1888 A	1886 A	1C40 A	1C4D A	1C8A A	1C8B A	1C8C A	1C8D A	3130 A
VCBUS	VCCO	VCCP	VCDS	VCDHK	VCDT	VCER	VCER1	VCER2	VCER3
1D92 A	1D15 A	1D14 A	3151 A	1C92 A	312F A	1DA2 A	1C72 A	1C76 A	1C7A A
VCER4	VCGTF	VCHD	VCHE	VCHFE	VCK	VCK0	VCK1	VCK2	VCK21
1C7E A	1C6B A	1C82 A	1CEC A	1CBC A	1CF6 A	1D0C A	1D10 A	1D26 A	1D27 A
VCK3	VCK5	VCK6	VCK7	VCK8	VCK9	VCKFE	VCLAS	VCLD	VCLOG
1D33 A	1D42 A	1D5A A	1D5E A	1D61 A	1D77 A	1CED A	1CL9 A	1C83 A	3165 A
VCM15	VCMA	VCMOF	VCMT	VCNK	VCRET	VCSDB	VCST1	VCST2	VCST3
1D22 A	1C84 A	1C86 A	1C85 A	1D8D A	1C3E A	3000 A	1C87 A	1C88 A	1C89 A

```
VCVC   VCVNB  XBUF   XCCE   XCER   XCFT   XCFTE  XCIWC  XCIWP  XERR
3131 A 1D9B A 3119 A 1F66 A 1F43 A 1F85 A 1F96 A 311C A 311D A 32BF A

XERR0  XERR1  XERRF  XFO1   XFO2   XFOW   XIMX   XLOF1  XLOF2  XLON1
32C0 A 32C2 A 32BE A 1F5C A 1F72 A 1F4B A 1F96 A 1F9A A 1FA0 A 1F97 A

XLON2  XMCE   XOCW1  XOCW2  XRTC   XSDC   XSDW   XWDT
1F9D A 1F50 A 311A A 311B A 1F28 A 1F36 A 3119 A 1F32 A

   1BC0    7F96
```

SNUMB = PB14C, ACTIVITY # = 03, REPORT CODE = 74, RECORD COUNT = 000026

```
ORIGIN   DATE MODULE ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION   ENTRY LOCATION

                 SUBPROGRAMS INCLUDED IN DECK.

            $     OPTION  FORTRAN,NOMAP
            $     LIBRARY L1

                 SUBPROGRAMS OBTAINED FROM USERS  LIBRARY   L1

                 SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

                                 RANGE              SIZE
            ALLOCATED CORE    000000 THRU 047777    050000
               RELOCATABLE    003400 THRU 047777    044400
            $      PRMFL   L1,R,R,USERLIB/UTILITY/BUTIL
            $      FILE    10,X1R
            $      FILE    11,X2R,40L
            $      FILE    S1,X3R,44R

            FCB AND BUFFER SPACE
               AVAILABLE      000101 THRU 003377    003277
               FILE CTRL BLKS 003174 THRU 003400    000205
               MAXIMUM BUFFER SPACE REQUIRED        002404

                20K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY WITH ALL FILES OPEN     730517 F/8
            001306 LOCATIONS REQUIRED FOR LOAD TABLE
            EXECUTION PROGRAM ENTERED AT  047356   THROUGH  .FSETU
```

SNUMB = PB14C, ACTIVITY # = 03, REPORT CODE = 06, RECORD COUNT = 001969

220	ABBUF										
		1084	1108	3528	4136						
221	ABDIN										
		1062	1233	1252	1283	1435	1443	2922	3013		
222	ABERR										
		1415									
394	ABLSMU										
		NOT USED									
179	ABPAN										
		1694	1904								
343	ABPD1										
		2006	2073	2106	2139	2172	2207				
344	ABPD2										
		2246	2313	2346	2379	2412	2447				
345	ABPD3										
		2486	2553	2586	2619	2652	2687				
342	ABPPD										
		1866	3059	3365	4147	4162					
175	ABRTC										
		505	837								
8	ACO										
		437	454	455	460	475	478	480	490	491	498
		499	537	561	565	567	568	585	586	588	589
		591	592	599	600	601	602	603	604	605	606
		607	608	609	610	614	619	638	655	659	660
		661	663	664	666	669	670	686	687	690	711
		712	714	718	742	743	748	752	762	763	765
		766	768	789	790	791	792	801	802	803	804
		827	853	854	871	876	881	883	884	901	902
		904	905	907	908	914	919	924	943	947	949
		950	967	968	970	971	973	974	979	994	999
		1020	1021	1023	1027	1062	1077	1078	1082	1083	1086
		1087	1093	1094	1096	1097	1099	1100	1101	1106	1107
		1110	1111	1117	1118	1120	1121	1123	1124	1125	1128
		1141	1144	1146	1148	1150	1152	1155	1156	1157	1158
		1160	1161	1163	1168	1169	1175	1176	1182	1183	1191
		1192	1195	1196	1197	1198	1202	1203	1206	1207	1214
		1220	1233	1234	1238	1241	1243	1244	1246	1247	1252
		1253	1257	1260	1262	1263	1265	1266	1283	1288	1289
		1291	1292	1294	1295	1297	1307	1308	1314	1315	1339
		1340	1341	1343	1352	1353	1358	1359	1378	1380	1382
		1383	1384	1387	1389	1391	1394	1396	1398	1401	1403
		1405	1410	1411	1413	1415	1417	1419	1422	1424	1427
		1428	1430	1432	1435	1438	1440	1443	1446	1448	1467
		1468	1486	1489	1490	1491	1492	1496	1497	1498	1499
		1500	1501	1502	1503	1504	1516	1517	1538	1539	1541
		1542	1544	1549	1550	1552	1553	1555	1560	1561	1563
		1564	1566	1571	1572	1574	1575	1577	1582	1583	1585
		1586	1588	1593	1594	1596	1597	1599	1604	1605	1607
		1608	1610	1615	1616	1618	1619	1621	1626	1627	1629
		1630	1632	1637	1638	1640	1641	1643	1648	1649	1651
		1652	1654	1659	1660	1662	1663	1665	1670	1671	1673
		1674	1676	1697	1698	1699	1700	1701	1738	1757	1765
		1773	1781	1789	1804	1809	1814	1819	1841	1844	1866
		1867	1870	1873	1876	1877	1878	1880	1882	1893	1894
		1895	1896	1897	1922	1944	1965	1980	2007	2019	2020
		2030	2031	2032	2034	2038	2039	2041	2043	2046	2047

2053	2068	2069	2070	2074	2085	2086	2101	2102	2103
2	21	211	134	5		21	215	152	57
2168	2169	2173	2180	2187	2202	2203	2204	2208	2220
2221	2232	2233	2234	2247	2259	2260	2270	2271	2272
2274	2278	2279	2281	2283	2286	2287	2293	2308	2309
2310	2314	2325	2326	2341	2342	2343	2347	2358	2359
2374	2375	2376	2380	2391	2392	2407	2408	2409	2413
2420	2427	2442	2443	2444	2448	2460	2461	2472	2473
2474	2487	2499	2500	2510	2511	2512	2514	2518	2519
2521	2523	2526	2527	2533	2548	2549	2550	2554	2565
2566	2581	2582	2583	2587	2598	2599	2614	2615	2616
2620	2631	2632	2647	2648	2649	2653	2660	2667	2682
2683	2684	2688	2700	2701	2712	2713	2714	2805	2806
2807	2809	2810	2812	2813	2815	2816	2818	2819	2836
2849	2860	2894	2896	2902	2904	2922	2927	2928	2963
2964	2966	2968	2987	2988	2997	3003	3013	3016	3020
3021	3026	3027	3028	3029	3031	3032	3034	3048	3050
3051	3054	3059	3091	3092	3095	3097	3105	3106	3108
3109	3116	3124	3125	3126	3128	3129	3131	3187	3188
3328	3330	3332	3334	3336	3339	3341	3344	3345	3348
3350	3354	3355	3357	3365	3366	3372	3374	3375	3380
3390	3391	3395	3398	3399	3400	3403	3404	3407	3408
3415	3416	3417	3419	3420	3436	3445	3446	3450	3451
3453	3455	3456	3459	3462	3463	3466	3468	3469	3482
3483	3485	3497	3498	3500	3504	3505	3506	3511	3512
3513	3528	3591	3598	3599	3600	3601	3602	3603	3604
3605	3606	3607	3608	3609	3610	3611	3613	3621	3624
3627	3628	3635	3651	3652	3655	3656	3658	3660	3662
3663	3665	3667	3669	3672	3673	3674	3677	3738	3739
3741	3742	3744	3745	3748	3749	3750	3751	3755	3756
3776	3777	3806	3813	3830	3837	3864	3865	3866	3871
3876	3879	3881	3931	3939	3940	3943	3946	3956	3957
3958	3963	3969	3972	3974	3995	3998	4005	4013	4016
4024	4035	4036	4042	4051	4063	4076	4077	4078	4079
4098	4110	4119	4138	4141	4143	4147	4182	4184	4186
4188	4190	4192	4194	4206	4208	4211	4216	4226	4228
4232	4233	4234	4235	4236	4237	4241	4242	4243	4244
4246	4250	4251	4253	4258	4260	4261	4262	4264	4265
4267	4271	4272	4274	4276	4277	4279	4284	4286	4287
4288	4290	4291	4293	4297	4298	4301	4305	4306	4307
4308	4309	4310	4380	4386	4392	4398			

9 AC1

519	520	526	527	562	570	581	615	624	634
639	697	703	719	749	754	872	886	897	944
952	963	1006	1017	1028	1085	1087	1092	1109	1111
1116	1143	1144	1147	1148	1151	1152	1236	1239	1241
1255	1258	1260	1321	1322	1323	1328	1346	1384	1389
1391	1396	1398	1403	1405	1422	1424	1430	1432	1438
1440	1446	1448	1450	1511	1512	1514	1516	1519	1520
1521	1537	1548	1559	1570	1581	1592	1603	1614	1625
1636	1647	1658	1669	1720	1726	1732	1736	1737	1744
1755	1756	1763	1764	1771	1772	1779	1780	1787	1788
1840	1844	1887	1888	1906	1907	1910	1911	1927	1928
1931	1932	1949	1950	1953	1954	1970	1971	1974	1975
1986	1987	1989	1990	2011	2013	2021	2023	2027	2032
2051	2057	2060	2061	2078	2079	2087	2091	2092	2095
2111	2112	2120	2124	2125	2128	2144	2145	2153	2157
2158	2161	2177	2184	2193	2194	2195	2196	2212	2214
2222	2226	2251	2253	2261	2263	2267	2272	2291	2297
2300	2301	2318	2319	2327	2331	2332	2335	2351	2352
2360	2364	2365	2368	2384	2385	2393	2397	2398	2401
2417	2424	2433	2434	2435	2436	2452	2454	2462	2466
2491	2493	2501	2503	2507	2512	2531	2537	2540	2541
2558	2559	2567	2571	2572	2575	2591	2592	2600	2604
2605	2608	2624	2625	2633	2637	2638	2641	2657	2664
2673	2674	2675	2676	2692	2694	2702	2706	2730	2731
2735	2736	2741	2742	2746	2747	2752	2753	2757	2758
2763	2764	2767	2768	2771	2772	2776	2778	2781	2782
2785	2786	2793	2794	2798	2836	2837	2838	2841	2842
2843	2844	2845	2846	2849	2850	2851	2854	2855	2856
2857	2860	2861	2862	2865	2893	2894	2895	2896	2901
2902	2903	2904	2917	2919	2930	2952	2953	2954	2961
2983	2985	2986	2992	2999	3000	3009	3067	3068	3075
3076	3083	3084	3111	3132	3133	3135	3138	3142	3144
3148	3151	3186	3189	3384	3385	3412	3422	3472	3473
3475	3476	3486	3487	3490	3491	3501	3502	3511	3516
3517	3534	3535	3538	3539	3541	3544	3549	3550	3553
3554	3556	3559	3615	3616	3618	3632	3645	3646	3686
3692	3698	3705	3753	3754	3767	3769	3770	3773	3774
3775	3788	3804	3807	3809	3822	3824	3828	3831	3833
3846	3848	3869	3872	3874	3877	3884	3885	3887	3888
3889	3891	3892	3895	3896	3929	3938	3944	3954	3955
3960	3961	3964	3966	3967	3970	3977	3978	3980	3981
3982	3984	3985	3988	3989	3992	3993	3996	4014	4023
4025	4043	4044	4060						

10 AC2

464	466	474	475	479	480	481	483	488	490
492	494	497	499	513	514	518	519	524	525
526	559	560	561	562	563	571	573	575	577
579	580	612	613	614	615	616	623	626	628
630	632	633	637	638	639	640	641	654	655
658	659	660	663	664	666	670	685	687	698
699	700	702	704	706	717	718	719	720	721
750	752	753	757	758	759	760	763	766	852
855	869	870	871	872	873	887	889	891	893
895	896	941	942	943	944	945	953	955	957
959	961	962	1007	1008	1009	1011	1013	1015	1026
1027	1028	1029	1030	1081	1082	1084	1085	1091	1092

1093	1096	1097	1099	1105	1106	1108	1109	1115	1116
1[illegible]	11[illegible]	112[illegible]	[illegible].23	[illegible]4	[illegible]	11[illegible]	116[illegible]	[illegible]161	[illegible]53
1168	1169	1175	1176	1182	1183	1189	1191	1192	1196
1202	1203	1215	1220	1237	1240	1244	1256	1259	1263
1320	1323	1324	1328	1336	1337	1338	1339	1340	1343
1379	1380	1495	1496	1498	1499	1501	1502	1504	1510
1511	1512	1514	1517	1519	1521	1717	1723	1729	1735
1737	1738	1744	1754	1756	1757	1762	1764	1765	1770
1772	1773	1778	1780	1781	1786	1788	1789	1841	1845
1885	1888	1911	1932	1954	1975	2006	2007	2012	2013
2029	2030	2041	2054	2055	2073	2074	2106	2107	2139
2140	2172	2173	2179	2180	2188	2189	2190	2192	2193
2195	2196	2207	2208	2213	2214	2246	2247	2252	2253
2269	2270	2281	2294	2295	2313	2314	2346	2347	2379
2380	2412	2413	2419	2420	2428	2429	2430	2432	2433
2435	2436	2447	2448	2453	2454	2486	2487	2492	2493
2509	2510	2521	2534	2535	2553	2554	2586	2587	2619
2620	2652	2653	2659	2660	2668	2669	2670	2672	2673
2675	2676	2687	2688	2693	2694	2727	2728	2731	2733
2734	2736	2738	2739	2742	2744	2745	2747	2749	2750
2753	2755	2756	2758	2762	2763	2766	2767	2770	2771
2775	2776	2780	2781	2784	2785	2789	2794	2798	2806
2807	2810	2813	2816	2819	2835	2842	2844	2846	2848
2855	2857	2859	2865	2918	2919	2926	2951	2954	2955
2958	2959	2962	2964	2984	2985	2988	2996	3017	3019
3020	3025	3027	3028	3031	3032	3034	3049	3050	3051
3065	3068	3076	3084	3122	3124	3125	3128	3129	3131
3154	3159	3166	3170	3187	3192	3335	3336	3339	3341
3373	3374	3375	3383	3384	3385	3387	3402	3411	3422
3436	3449	3451	3452	3453	3455	3457	3461	3462	3465
3467	3468	3469	3474	3475	3479	3481	3482	3485	3489
3490	3494	3496	3497	3500	3504	3508	3510	3529	3544
3559	3587	3588	3589	3599	3601	3603	3605	3607	3609
3611	3621	3628	3652	3656	3658	3660	3663	3665	3667
3669	3673	3738	3739	3741	3742	3744	3745	3748	3749
3750	3751	3753	3757	3773	3774	3778	3805	3806	3807
3810	3812	3814	3829	3830	3831	3834	3836	3838	3865
3866	3870	3871	3872	3875	3876	3877	3879	3881	3888
3891	3892	3895	3929	3930	3932	3938	3940	3943	3944
3945	3947	3955	3958	3960	3961	3963	3964	3966	3967
3969	3970	3972	3974	3981	3984	3985	3988	3993	3995
3996	3997	3999	4006	4013	4014	4015	4017	4023	4026
4028	4099	4104	4115	4121	4136	4138	4141	4143	4162
4166	4171	4182	4184	4186	4188	4190	4192	4194	4205
4206	4208	4211	4225	4226	4231	4232	4234	4235	4237
4240	4241	4243	4244	4246	4257	4258	4259	4260	4261
4264	4265	4267	4276	4283	4284	4285	4286	4287	4290
4291	4293	4303	4307	4310					

11 AC3

436	459	465	470	471	472	489	491	493	538
542	543	563	572	573	574	575	576	577	578
579	616	618	625	626	627	628	629	630	631
632	640	673	674	675	679	680	700	701	702
703	704	705	706	707	720	826	873	875	888
889	890	891	892	893	894	895	920	923	945
954	955	956	957	958	959	960	961	978	993
1009	1010	1011	1012	1013	1014	1015	1016	1029	1216
1217	1235	1236	1237	1238	1239	1240	1243	1246	1254
1255	1256	1257	1258	1259	1262	1265	1312	1314	1320
1329	1330	1332	1347	1348	1350	1353	1354	1356	1537
1538	1541	1542	1544	1548	1549	1552	1553	1555	1559
1560	1563	1564	1566	1570	1571	1574	1575	1577	1581
1582	1585	1586	1588	1592	1593	1596	1597	1599	1603
1604	1607	1608	1610	1614	1615	1618	1619	1621	1625
1626	1629	1630	1632	1636	1637	1640	1641	1643	1647
1648	1651	1652	1654	1658	1659	1662	1663	1665	1669
1670	1673	1674	1676	1704	1707	1710	1713	1719	1720
1725	1726	1731	1732	1740	1759	1767	1775	1783	1791
1806	1811	1816	1821	1828	1901	1916	1924	1938	1946
1959	1967	1982	2010	2011	2017	2022	2023	2026	2027
2034	2039	2042	2043	2044	2050	2051	2056	2057	2061
2066	2077	2078	2079	2083	2091	2094	2095	2099	2110
2111	2112	2116	2124	2127	2128	2132	2143	2144	2145
2149	2157	2160	2161	2165	2176	2177	2200	2211	2212
2218	2225	2226	2230	2250	2251	2257	2262	2263	2266
2267	2274	2279	2282	2283	2284	2290	2291	2296	2297
2301	2306	2317	2318	2319	2323	2331	2334	2335	2339
2350	2351	2352	2356	2364	2367	2368	2372	2383	2384
2385	2389	2397	2400	2401	2405	2416	2417	2440	2451
2452	2458	2465	2466	2470	2490	2491	2497	2502	2503
2506	2507	2514	2519	2522	2523	2524	2530	2531	2536
2537	2541	2546	2557	2558	2559	2563	2571	2574	2575
2579	2590	2591	2592	2596	2604	2607	2608	2612	2623
2624	2625	2629	2637	2640	2641	2645	2656	2657	2680
2691	2692	2698	2705	2706	2710	2729	2730	2735	2740
2741	2746	2751	2752	2757	2761	2764	2768	2772	2777
2778	2782	2786	2790	2795	2799	2805	2809	2812	2815
2818	2925	2926	2927	2934	2935	2936	2937	2948	2949
2955	2959	2991	2992	2995	2996	2997	3002	3003	3008
3009	3018	3019	3103	3147	3148	3152	3153	3154	3155
3157	3158	3159	3160	3165	3168	3169	3170	3171	3188
3193	3363	3364	3366	3387	3389	3391	3392	3395	3400
3402	3404	3405	3408	3423	3442	3443	3445	3448	3449
3450	3456	3464	3465	3466	3473	3478	3479	3483	3487
3493	3494	3498	3502	3505	3509	3510	3512	3513	3515
3516	3578	3580	3581	3584	3586	3588	3591	3598	3600
3602	3604	3606	3608	3610	3613	3616	3619	3633	3636
3637	3639	3644	3646	3647	3648	3678	3679	3681	3687
3693	3699	3704	3797	3798	3800	3858	3859	3861	3863
3869	3874	3962	3968	4008	4025	4032	4036	4045	4046
4048	4052	4053	4055	4061	4100	4102	4102	4103	4104
4107	4108	4108	4115	4120	4121	4135	4139	4145	4163
4168	4173	4204	4210	4215	4270	4271	4272	4273	4274
4296	4297	4298	4300	4301	4304	4305	4306	4308	

```
402  ACCU
               NOT USED
774  ACCUI
                742
776  ACDISP
                769
395  ACMU
                459
176  ACOTF
                834
777  ACRAM
                765
321  ACSAV
               NOT USED
322  ACSV2
                941   942  1026  1030
323  ACSVF
                950   953  1020
324  ACSVS
                971   974   999  1007
336  ACTST
               3049
778  ACWAP
                750   757
385  ADEND
                483   789  4076
398  ADMU
               NOT USED
364  AE100
               2189  2429  2669
365  AE10
               2054  2294  2534
162  AEADC
                643   727   813  1066  1748  3036
151  AEBKD
                542
291  AEBUF
                474   654   685
149  AECCU
                444
779  AECCUR
                762
152  AECER
                667
299  AECMF
               1078  1101  1125  1128  4228
155  AECOM
                839
306  AED23
               1247  1379  3116
310  AEDERR
               1338
```

161	AEDET										
		843									
309	AEDETT										
		1329	1332	1336	1346	1347	1350				
154	AEDIS										
		838									
308	AEDMIT										
		1294	1297	1307	1359						
307	AEDNO										
		1235									
332	AEDSV2										
		869	870								
333	AEDSVF										
		884	887								
334	AEDSVS										
		905	908								
159	AEEOC										
		3001									
297	AEERR										
		658	1091	1115	1154	1330	1348	1410	1427		
148	AEGIS										
		139									
189	AEM10										
		2025	2093	2126	2159	2224	2265	2333	2366	2399	2464
		2505	2573	2606	2639	2704					
190	AEM24										
		2052	2292	2532							
150	AEMAD										
		448									
157	AEMIC										
		841									
367	AEP1E										
		1698	1804	1894	1922	2020	2047	2069	2086	2102	2119
		2135	2152	2168	2203	2221	2233				
368	AEP2E										
		1699	1809	1895	1944	2260	2287	2309	2326	2342	2359
		2375	2392	2408	2443	2461	2473				
369	AEP3E										
		1700	1814	1896	1965	2500	2527	2549	2566	2582	2599
		2615	2632	2648	2683	2701	2713				
180	AEPAN										
		502									
186	AEPBI2										
		2174	2414	2654							
187	AEPBI3										
		2008	2075	2108	2141	2209	2248	2315	2348	2381	2449
		2488	2555	2588	2621	2689					
351	AEPCD1										
		2026	2042	2056	2094	2127	2160	2179	2192	2225	2729
		2762	2775	2790							
352	AEPCD2										
		2266	2282	2296	2334	2367	2400	2419	2432	2465	2740
		2766	2780	2795							

353	AEPCD3										
		2506	2522	2536	2574	2607	2640	2659	2672	2705	2751
		2770	2784	2799							
358	AEPCF										
		1877	1882	1885	1907	1928	1950	1971	1990		
362	AEPCY										
		2012	2021	2252	2261	2492	2501				
347	AEPD1										
		2010	2022	2050	2077	2110	2143	2176	2211		
348	AEPD2										
		2250	2262	2290	2317	2350	2383	2416	2451		
349	AEPD3										
		2490	2502	2530	2557	2590	2623	2656	2691		
296	AEPDT										
		NOT USED									
370	AEPDVE										
		1701	1819	1897	1980	2070	2103	2136	2169	2204	2234
		2310	2343	2376	2409	2444	2474	2550	2583	2616	2649
		2684	2714								
372	AEPEF1										
		1740	1759	2017	2044	2066	2083	2099	2116	2132	2149
		2165	2200	2218	2230						
373	AEPEF2										
		1767	1775	2257	2284	2306	2323	2339	2356	2372	2389
		2405	2440	2458	2470						
374	AEPEF3										
		1783	1791	2497	2524	2546	2563	2579	2596	2612	2629
		2645	2680	2698	2710						
355	AEPFT										
		1719	1725	1731	2029	2269	2509				
363	AEPIN1										
		2213	2222	2453	2462	2693	2702				
359	AEPIR										
		1873	1880								
182	AEPP1										
		1795	1919								
183	AEPP2										
		1798	1941								
184	AEPP3										
		1801	1962								
181	AEPPM										
		848									
295	AEPST										
		1198	1207	1387	2962						
357	AEPUF										
		1870	1987	3097	3106						
185	AEPUP										
		1826	3101								
303	AER123										
		1141	1146	1150	1155	1266	1378				
311	AERAM										
		455	986								

192	AERE0						
		1745	4062				
202	AERE10						
		2117	2357	2597			
203	AERE11						
		2150	2390	2630			
204	AERE12						
		2219	2459	2699			
193	AERE1						
		2067	2307	2547	3688		
194	AERE2						
		2100	2340	2580	3694		
195	AERE3						
		2133	2373	2613	3700		
196	AERE4						
		1349	2166	2406	2646	3706	
197	AERE5						
		1331	2201	2441	2681	3634	
198	AERE6						
		2231	2471	2711	3620		
199	AERE7						
		3623					
200	AERE8						
		2018	2045	2258	2285	2498	2525
201	AERE9						
		2084	2324	2564	3424		
304	AERNO						
		1254					
147	AERSI						
		136					
316	AESAV						
		NOT USED					
153	AESCH						
		677					
160	AESDW						
		847	912				
317	AESV2						
		559	560	717	721		
318	AESVF						
		568	571	711			
319	AESVS						
		589	592	690	698		
301	AESYD						
		1215	2789				
156	AESYN						
		840					
293	AETCY						
		497	1288	1291	1510		
158	AETIM						
		842					
294	AETST						
		1292	2958				

327	AISTK1	600
328	AISTK2	603
329	AISTK3	606
330	AISTK4	609 623
326	AISV2	612 613 637 641
393	ALDU	4139 4204 4215
231	APBUF	NOT USED
167	APCIM	NOT USED
240	APEMAX	2777 3443 3446 3452
241	APERR	1419 3423 3619 3633 3687 3693 3699 3704 4061
238	APFREQ	1486 1492 3417
232	APIP	1394 3054 3095 3345 3380 3416
239	APMDT	NOT USED
237	APNIF	1495
163	APMON	844
406	APP1U	1704 1806 1916 1924 4163
407	APP2U	1707 1811 1938 1946 4168
408	APP3U	1710 1816 1959 1967 4173
405	APPDU	1713 1821 1828 1901 1982 3103 4145
165	APREQ	NOT USED
3434	APSWCH	3326
242	APTAVT	NOT USED
234	APTCP	NOT USED
235	APTCS	NOT USED
236	APTEI	NOT USED
233	APTEM	3398 3407 3463 3467 3481 3496 3506

```
166  APTER
            NOT USED
243  APTITE
            NOT USED
245  APTPMT
            NOT USED
400  ARTC
              436   538   618   826   875   978   993
399  ASDB
            NOT USED
397  ASDU
              923  4210
381  ASEAD
              792   804  4079  4099
212  ASMAD
              794   806  4081
380  ASSAD
              790   802  4077  4100  4107
279  ATAI1
             1736  1763  1779  2727  2738  2749
280  ATAI2
             1755  1771  1787  2733  2744  2755
256  ATBUF
             4225  4257  4283  4303
258  ATCDS
             1216  2952  3065  3111  3132  3147  3151  3152  3157  3168
             3363  3442
284  ATCIW
              525  2925  3017
257  ATCIWC
              527  2928  3021
169  ATCOPY
             3156  3161  3167  3172
396  ATCP
              920  4135
271  ATCTP
              518  2995  3002  3155  3165  3448  3464  3478  3493  3509
171  ATDIAL
              510
264  ATDWL
             2761  2951
282  ATECI
             3335  3373  3383  3457  3461  3474  3489  3508
263  ATEMP
            NOT USED
262  ATEOC
            NOT USED
272  ATERR
             1417  3025  3122
261  ATFST
             1468  1490  2948  2966  2968  3092  3109  3420
```

```
 278  ATHI
                513   524  2918  3008
 281  ATIMX
               2991
 259  ATIN
               514  2917  2984  2986  3000  3018  3026  3334  3372
 260  ATINT
               520  2934  2936  3016  3515  3517
 170  ATNCY
              1477
 270  ATNTP
              3160  3166  3171  3392  3405
 287  ATPE
               494
 168  ATPEX
               845
 267  ATPM1
               489  1717  1735  1754  2728  2734
 268  ATPM2
              1723  1762  1770  2739  2745
 269  ATPM3
              1729  1778  1786  2750  2756
 265  ATPP
              3153  3158  3169
 286  ATPTAB
               488
 277  ATSCI
            NOT USED
 247  AVCBT
              1401  3328  3332  4044
3922  AVCBUS
              3849
 251  AVCDB
              3589
 250  AVCDT
              3350  3354  3578  3618  3632  3645  3686  3692  3698  3705
              4060
3924  AVCER
              3937  3952  4004  4022
4038  AVCK
              4037
 164  AVCLAS
              3351
 249  AVCLOG
              3636  3678  3797  3858  4045  4052
 252  AVCSDB
               479  3529  3587
 248  AVCVC
              1312  1321  3647
3923  AVCVNB
              3820  3823  3826  3844  3847  3850
```

226	AXBUF										
		914	919	924	1081	1105	1450	4205	4231	4240	4273
		4300	4304								
4323	AXCFT										
		4270	4296								
227	AXERR										
		1413	4259	4285							
207	AXLOF1										
		1705	1708	1711	1829	1917	1939	1960	3104		
209	AXLOF2										
		1714	1902								
206	AXLON1										
		1807	1812	1817	1925	1947	1968				
208	AXLON2										
		1822	1983								
172	AXRTC										
		836									
174	AXSDC										
		530	846								
173	AXWDT										
		1131	1746								
4429	BBUF										
		220									
4431	BCP1										
		NOT USED									
4432	BCR2										
		NOT USED									
4430	BDIN										
		221									
4677	BERR										
		222									
4678	BERRF										
		NOT USED									
4162	BPAN										
		179									
4463	BPD1										
		343									
4464	BPD2										
		344									
4465	BPD3										
		345									
4181	BPIN										
		4164	4169	4174							
4452	BPPD										
		342									
4135	BRTC										
		175									
91	CO										
		437	460	478	588	619	686	748	768	827	876
		904	970	979	994	1077	1195	1206	1352	1358	1489
		1491	1520	1697	1840	1893	1989	2038	2053	2184	2187
		2278	2293	2424	2427	2518	2533	2664	2667	2835	2848
		2859	2963	2987	2999	3048	3091	3105	3108	3344	3399
		3419	3459	3472	3486	3501	3534	3549	3615	3651	3755
		3776	3864	3939	3956	3957	3992	4005	4024	4042	4216
		4309	4380	4386	4392	4398					

102	C125										
		3627									
100	C15										
		NOT USED									
92	C1										
		466	481	492	493	537	585	591	661	669	753
		855	901	907	967	973	1094	1100	1118	1158	1197
		1289	1295	1322	1341	1354	1467	1539	1550	1561	1572
		1583	1594	1605	1616	1627	1638	1649	1660	1671	1845
		1906	1910	1986	2019	2031	2046	2068	2085	2092	2101
		2118	2125	2134	2151	2158	2167	2194	2202	2220	2232
		2259	2271	2296	2308	2325	2332	2341	2358	2365	2374
		2391	2398	2407	2434	2442	2460	2472	2499	2511	2526
		2548	2565	2572	2581	2598	2605	2614	2631	2638	2647
		2674	2682	2700	2712	2961	2983	3029	3067	3126	3144
		3189	3192	3193	3348	3355	3415	3538	3553	3624	3635
		3677	3954	4051	4063	4103	4110	4120	4262	4288	
93	C2										
		1124	1143	1927	1931	3075	3138				
94	C3										
		624	1949	1953	3083	3142					
101	C48										
		749									
95	C4										
		570	633	697	886	952	1006	1147	1970	1974	
96	C5										
		NOT USED									
97	C6										
		1214	2793	4035							
98	C7										
		4166	4171								
99	C8										
		1151									
108	CCBO										
		587	903	969	4111						
109	CCB1										
		NOT USED									
116	CCCF										
		832									
113	CCCU										
		739									
106	CCEZ										
		656	691	1000	1088	1112	1142	1145	1149	1153	1170
		1177	1184	1388	1395	1402	1420	1429	1739	1758	1766
		1774	1782	1790	1805	1810	1815	1820	1842	1883	1923
		1945	1966	1981	3329	3592	4252	4278			
110	CCNZ										
		1129	1242	1245	1261	1264	1412	1414	1416	1418	1487
		1868	1874	3004	3055	3096	3098	3352	3381	3617	3622
		4227	4229								
114	CCPH										
		534									

111	CCSF										
		435	557								
112	CCZN										
		1843	2033	2181	2273	2421	2513	2661	3396		
107	CCZP										
		1518	2998	3458	3484	3499	4010	4030			
84	CM100										
		NOT USED									
87	CM10										
		2087	2120	2153	2327	2360	2393	2567	2600	2633	
83	CM110										
		NOT USED									
86	CM16										
		464									
90	CM1										
		470	581	634	673	708	754	897	963	1017	1217
		1876	1887	2953	3330	3411	3412	3584	3644	4098	4119
85	CM20										
		3186									
89	CM2										
		NOT USED									
88	CM6										
		766									
954	COTF1										
		964									
973	COTF2										
		969									
977	COTF3										
		972									
1003	COTF4										
		1000									
1008	COTF5										
		1018									
940	COTF										
		176									
4590	CSAV										
		321									
4593	CSV2										
		322									
4595	CSVF										
		323									
4597	CSVS										
		324									
4622	CTST										
		336									
46	DO										
		475	545	655	664	665	666	670	687	706	771
		818	858	918	1015	1032	1064	1097	1099	1121	1123
		1133	1161	1163	1164	1191	1210	1223	1236	1238	1243
		1246	1255	1257	1262	1265	1269	1299	1340	1343	1361
		1452	1482	1496	1498	1506	1511	1521	1523	1542	1543
		1544	1545	1553	1555	1556	1564	1566	1567	1575	1577
		1578	1586	1588	1589	1597	1599	1600	1608	1610	1611
		1619	1621	1622	1630	1632	1633	1641	1643	1644	1652
		1654	1655	1663	1665	1666	1674	1676	1677	1720	1738
		1757	1765	1773	1781	1789	1831	1841	1849	1992	2007
		2011	2013	2023	2030	2041	2051	2055	2079	2091	2112

2124	2145	2157	2190	2193	2214	2236	2247	2251	2253
2[illegible]	22[illegible]	229[illegible]	[illegible]95	[illegible]9	[illegible]	22[illegible]	236[illegible]	[illegible]385	[illegible]97
2430	2433	2454	2476	2487	2491	2493	2503	2521	2531
2535	2559	2571	2592	2604	2625	2637	2670	2673	2694
2716	2730	2741	2752	2764	2802	2806	2821	2868	2897
2905	2929	2933	2939	2969	3022	3032	3034	3050	3129
3130	3131	3154	3159	3162	3170	3173	3191	3195	3360
3366	3368	3395	3400	3408	3421	3425	3439	3445	3450
3453	3455	3456	3466	3468	3469	3473	3482	3483	3485
3487	3497	3498	3500	3502	3504	3505	3512	3513	3516
3518	3551	3555	3560	3600	3601	3625	3628	3638	3639
3641	3648	3663	3673	3680	3683	3744	3748	4047	4050
4054	4057	4064	4117	4138	4148	4176	4196	4211	4213
4218	4226	4238	4247	4258	4265	4266	4267	4268	4276
4284	4291	4293	4294	4307	4310	4311	4382	4388	4394
4400									

56	D10										
		1390	1581	1653	3876	3945	3969	3995	4015	4055	
73	D116										
		854									
74	D118										
		853									
57	D11										
		1593	1596	1664	3060	3877	3891	3929	3970	3984	3996
		4053									
58	D12										
		759	1592	1675	3531	3865	3940	3958	3993	4006	4028
59	D13										
		1604	1607	3546	3938	3955					
60	D14										
		760	1284	1603	2923	3014	3761	3782	3818	3842	3935
		3950	4002	4020							
61	D15										
		1063	1444	1615	1618	2967	3544	3866	4026	4245	4299
62	D16										
		1614	4023								
63	D17										
		586	902	968	1626	1629					
64	D18										
		1625									
65	D19										
		1637	1640								
47	D1										
		563	573	616	626	640	660	663	704	720	873
		889	915	945	955	1013	1029	1082	1085	1098	1168
		1237	1239	1256	1258	1337	1339	1447	1499	1501	1512
		1514	1538	1541	1554	1726	2034	2039	2074	2078	2180
		2188	2195	2196	2270	2274	2279	2314	2318	2420	2428
		2435	2436	2514	2519	2554	2558	2660	2668	2675	2676
		2735	2746	2757	2768	2778	2810	2843	2854	2867	2893
		2903	2965	3033	3051	3071	3125	3128	3339	3341	3374
		3375	3602	3603	3637	3656	3814	3838	3932	3947	3962
		4123	4141	4182	4206	4232	4234	4241	4243	4261	4264
		4274	4292								
66	D20										
		1636									
67	D21										
		1648	1651								
68	D22										
		1647									
69	D23										
		1659	1662								
70	D24										
		1658									
71	D25										
		1670	1673								

72	D26										
		1669									
48	D2										
		499	575	628	659	702	891	957	1011	1106	1109
		1122	1175	1240	1259	1439	1502	1504	1517	1519	1537
		1565	1732	2107	2111	2226	2347	2351	2466	2510	2587
		2591	2706	2772	2776	2781	2782	2785	2813	2841	2856
		2895	3124	3336	3436	3449	3510	3604	3605	3665	3681
		3739	3750	3753	3774	3961	3968	4143	4184	4208	4235
		4237	4244	4246	4260	4301					
49	D3										
		577	630	700	763	893	959	1009	1093	1096	1162
		1182	1549	1552	1576	2027	2043	2060	2140	2144	2267
		2283	2300	2380	2384	2507	2523	2540	2620	2624	2786
		2805	2816	2845	2901	3028	3031	3451	3462	3586	3606
		3607	3658	3679	3805	3829	3874	3875	4186	4287	4290
		4305	4308								
50	D4										
		447	579	580	632	895	896	961	962	1045	1092
		1202	1431	1548	1587	2095	2173	2177	2335	2413	2417
		2575	2653	2657	2809	2819	3027	3117	3465	3608	3609
		3667	3738	3742	3754	3775	3800	3869	3960	3967	4086
		4116	4188	4286	4306						
51	D5										
		450	1050	1117	1120	1169	1176	1183	1189	1192	1196
		1203	1423	1560	1563	1598	2128	2208	2212	2368	2448
		2452	2608	2688	2692	2807	2812	3079	3119	3140	3475
		3479	3598	3599	3621	3652	3660	3798	3810	3834	3870
		4008	4190								
52	D6										
		451	1054	1116	1382	1404	1436	1559	1609	2161	2401
		2641	2815	3490	3494	3610	3611	3669	3741	3745	3749
		3773	3861	3966	4025	4192					
53	D7										
		452	1058	1157	1160	1397	1571	1574	1620	2061	2301
		2541	2818	3384	3387	3402	3559	3591	3613	3616	3751
		3757	3778	3859	3999	4017	4194				
54	D8										
		1156	1570	1631	2057	2297	2537	2763	2767	2771	3389
		3806	3812	3830	3836	3871	3879	3881	3930	3943	3963
		3972	3974	3997	4013	4048	4251				
55	D9										
		1582	1585	1642	3087	3807	3831	3872	3888	3892	3895
		3944	3964	3981	3985	3988	4014	4046			
4771	DEND										
		385									
76	DETS										
		1367	3433	3712							
41	DM12										
		752									
45	DM1										
		562	615	639	719	872	944	1028			

44	DM2	561	614	638	718	871	943	1027
43	DM4	699	1008					
42	DM7	758						
78	DPMD	4043						
1855	E100	364						
1854	E10	365						
888	EADC1	898						
907	EADC2	903						
910	EADC3	906						
923	EADC4	917						
868	EADC	162						
4076	EBKD1	4087						
4073	EBKD	151						
4091	EBKDE	4082	4088					
4538	EBUF	291						
752	ECCU1	755						
736	ECCU	149						
775	ECCUI	743						
771	ECCUR	740	745	779				
726	ECER	152						
4549	ECMF	299						
1104	ECOM1	1088						
1128	ECOM2	1112						
1076	ECOM	155						
1133	ECOMR	1126	1129					
4567	ED23	306						

```
4766  EDERR
                310
1314  EDET1
               1357
1336  EDET2
               1325
1366  EDET32
               1324
1352  EDET3
               1317  1333
1306  EDET
                161
1364  EDETI
               1308
1365  EDETM
               1315
1361  EDETR
               1310
4573  EDETT
                309
1042  EDIS
                154
1066  EDISH
               1047  1052  1056  1060
 860  EDLAY
                852
4572  EDMIT
                308
4568  EDNO
                307
4612  EDSAV
            NOT USED
4615  EDSV2
                332
4617  EDSVF
                333
4619  EDSVS
                334
1466  EEOC
                159
4731  EERR0
            NOT USED
4732  EERR1
            NOT USED
4733  EERR2
            NOT USED
4734  EERR3
            NOT USED
4735  EERR4
            NOT USED
4736  EERR5
            NOT USED
```

```
4721  EERR
                297
4722  EERRF
            NOT USED
 572  EGIS1
                582
 591  EGIS2
                587
 649  EGIS3
                590   593
 694  EGIS4
                691
 699  EGIS5
                709
 556  EGIS
                148
 534  EHLT
            NOT USED
2892  EM10
                189
2900  EM24
                190
 785  EMAD
                150
 813  EMADE
                795   807
 818  EMADR
                811
1249  EMIC1
               1242  1245
1268  EMIC2
               1261  1264
1230  EMIC
                157
4653  EP1E
                367
4654  EP2E
                368
4655  EP3E
                369
1753  EPAN1
               1739
1691  EPAN
                180
1746  EPANE1
               1823
1743  EPANE
               1760  1768  1776  1784  1792
2659  EPBI1
            NOT USED
2848  EPBI2
                186
```

2835	EPBI3	187
2833	EPBI	NOT USED
2867	EPBIR	2840 2853 2864
4635	EPCD1	351
4645	EPCD2	352
4646	EPCD3	353
4659	EPCF	358
1852	EPCY	362
4630	EPD1	347
4631	EPD2	348
4632	EPD3	349
4547	EPDT	296
4656	EPDVE	370
4756	EPEF1	372
4759	EPEF2	373
4762	EPEF3	374
1485	EPEOC	1474
1841	EPFI1	1847
1839	EPFI	1718 1724 1730
1849	EPFIR	1843
4649	EPFT	355
1853	EPIN1	363
4660	EPIR	359
2065	EPP10	2009
2099	EPP11	2076
2132	EPP12	2109

2165	EPP13						
		2142					
2200	EPP14						
		2175					
2230	EPP15						
		2210					
2003	EPP1						
		182					
2305	EPP20						
		2249					
2339	EPP21						
		2316					
2372	EPP22						
		2349					
2405	EPP23						
		2382					
2440	EPP24						
		2415					
2470	EPP25						
		2450					
2243	EPP2						
		183					
2545	EPP30						
		2489					
2579	EPP31						
		2556					
2612	EPP32						
		2589					
2645	EPP33						
		2622					
2680	EPP34						
		2655					
2710	EPP35						
		2690					
2483	EPP3						
		184					
1880	EPPM1						
		1871					
1910	EPPM2						
		1890					
1931	EPPM3						
		1913					
1953	EPPM4						
		1934					
1974	EPPM5						
		1956					
1865	EPPM						
		181					
1992	EPPMR						
		1883	1908	1929	1951	1972	1977
4546	EPST						
		295					

4658	EPUF	357
2723	EPUP	185
2805	EPUPS	2791 2796 2800
4562	ER123	303
1536	ERE0	192
1646	ERE10	202
1657	ERE11	203
1668	ERE12	204
1547	ERE1	193
1558	ERE2	194
1569	ERE3	195
1580	ERE4	196
1591	ERE5	197
1602	ERE6	198
1613	ERE7	199
1624	ERE8	200
1635	ERE9	201
4563	ERN0	304
434	ERSI	147
669	ERTC1	656
654	ERTC	NOT USED
4539	ERTCF	NOT USED
858	ERTS	850
4580	ESAV	316
825	ESCH	153
1421	ESDW1	1412 1414 1416 1418

1375	ESDW	160				
625	ESTK1	635				
598	ESTK	435	557			
4583	ESV2	317				
4585	ESVF	318				
4587	ESV5	319				
1217	ESYA1	1221				
1222	ESYA2	1219				
1212	ESYA	1167	1174	1181	1187	
4550	ESYD	301				
1166	ESYN1	1145				
1173	ESYN2	1149				
1180	ESYN3	1153				
1188	ESYN4	1171	1178	1185		
1187	ESYN4A	1142				
1201	ESYN5	1194				
1209	ESYN6	1170	1177	1184	1199	1205
1140	ESYN	156				
490	ETCPY	495				
4543	ETCY	293				
1516	ETEOC1	1513				
1520	ETEOC2	1515	1518			
1509	ETEOC	1480				
1294	ETIM1	1286				
1280	ETIM	158				
1299	ETIMR	NOT USED				

4541	ETST						
		294					
1367	EVCVC						
		1356					
480	SZDB						
		484					
18	FC10						
		4191					
19	FC11						
		4181					
20	FC12						
		4183					
21	FC13						
		4185					
22	FC14						
		4187					
23	FC15						
		4189					
30	FC17						
		4193					
13	FC1						
		925	4207	4212			
14	FC2						
		4209					
15	FC3						
		4137	4140	4146			
32	FC41						
		439	461	828	878	981	4217
33	FC42						
		539	995				
34	FC43						
		438	620	877	921	980	
35	FC44						
		4381					
36	FC45						
		4387					
37	FC46						
		4393					
38	FC47						
		4399					
16	FC4						
		4142					
4314	H0006						
		NOT USED					
2876	H0009						
		2862					
1272	H000E						
		1253					
2873	H000F						
		2861					
1457	H0010						
		1428					

1273	H0030	1234				
4315	H007F	4277				
2875	H0090	2851				
2872	H00F0	2850				
421	H0200	454				
1995	H0400	1867	1878			
2874	H0900	2838				
2871	H0F00	2837				
549	H0FFF	498				
3727	H1T1	3822				
3728	H1T2	3824				
3729	H2T1	3846				
3730	H2T2	3848				
416	H3000	NOT USED				
422	H3BFF	4078				
417	H3FFF	791	1083	1086	1107	1110
4316	H7F00	4250				
415	H7FFF	1497	1500	1503	4233	4236 4242
419	HFC00	801				
420	HFCCF	803				
1455	HF800	1383				
1456	HFFEF	1411				
119	INEN	441	621	830	879	
4605	ISTK1	327				
4606	ISTK2	328				
4607	ISTK3	329				

4608	ISTK4								
		330							
4609	ISTK5								
		NOT USED							
4601	ISV0								
		NOT USED							
4602	ISV1								
		NOT USED							
4603	ISV2								
		326							
4604	ISV3								
		NOT USED							
4600	ISV								
		NOT USED							
4774	LAC								
		4							
4775	LBP								
		131	143	215	314	339	377	388	411
4783	LBX								
		4130	4153	4200	4222	4320	4372		
4784	LDB								
		4407	4577	4625	4663	4672			
4785	LDU								
		NOT USED							
4782	LEB								
		4070	4095						
4776	LFR								
		429	553	733	782	822	865	935	1527
4777	LFS								
		1039	1073	1137	1227	1277	1303	1372	1462
4778	LPP								
		1685	1859	2000	2240	2480	2720	2825	2880
4780	LPR								
		3324	3520						
4779	LTP								
		2912	2943	2978	3043	3177	3199		
4781	LVC								
		3572	3710	3733	3793	3853	3901	3927	4041
4476	PBUF								
		231							
127	PCIM								
		167							
4490	PEMAX								
		240							
4691	PERR								
		241							
4692	PERRF								
		NOT USED							
4483	PFREQ								
		238							
4477	PIP								
		232							

3349	PM01			
		3359		
3361	PM02			
		3353		
3386	PM05			
		3413		
3346	PM0			
		3338	3342	
3436	PM1			
		3378		
3440	PM2			
		3438		
3504	PM3			
		3470		
3505	PM4			
		3484	3499	
3507	PM5			
		3477	3492	3503
3431	PMAS2			
		3390	3403	
3427	PMAS			
		3364		
4484	PMDT			
		239		
3422	PMER10			
		3396		
3433	PMHD			
		3357		
4482	PMIF			
		237		
3432	PMM1			
		3476	3491	
3325	PMON			
		163		
3425	PMRET			
		3381		
126	PREQ			
		165		
3521	PSWCH			
		3434		
3563	PSWFC			
		3535	3539	3541
3564	PSWFD			
		3550	3554	3556
4485	PTAVT			
		242		
4479	PTCP			
		234		
4480	PTCS			
		235		
4481	PTEI			
		236		

4478	PTEM	233				
125	PTER	166				
4486	PTITE	243				
4487	PTPMT	245				
4667	SEAD	381				
120	SELF	692	694	1001	1003	1381
4102	SMAD1	4105				
4108	SMAD2	4112	4122			
4119	SMAD3	4109				
4098	SMAD	212				
4115	SMADE	4111				
4666	SSAD	380				
2971	T0	2930	2937	2949		
2972	T1	2935	3133			
2973	T2	3135				
2974	T3	NOT USED				
3213	TAI1	279				
3214	TAI2	280				
4519	TBUF	256				
4521	TCDS	258				
3241	TCIW	284				
4520	TCIWC	257				
3186	TCOPY	169				
4534	TCTP	271				
3144	TD1	3134	3141			
3138	TD2	3120	3136			

3142	TD3	3137
3151	TDCH	3112
3116	TDIAL	171
3146	TDSEL	3139 3143
4527	TDWL	264
3219	TECI	282
4526	TEMP	263
4525	TEOC	262
4717	TERR0	NOT USED
4718	TERR1	NOT USED
4713	TERR	272
4714	TERRF	NOT USED
3008	TEZ	NOT USED
3231	TFEP	3230
4524	TFST	261
3004	TGEZ	2998
3016	TGZ	3004
3212	THI	278
3215	TIMX	281
4522	TIN	259
4523	TINT	260
3233	TLEP	3230
3187	TLOOP	3194
2999	TLZ	NOT USED
2930	TMSG	2921 2924
3046	TNCY	170

3140	TND2				
		3118			
3165	TNDC				
		3055	3149		
4533	TNTP				
		270			
2981	TNXT				
		2938	3011	3015	
3023	TOL				
		2990	2994		
3317	TPE				
		287			
2915	TPEX				
		168			
4530	TPM1				
		267			
4531	TPM2				
		268			
4532	TPM3				
		269			
3263	TPTAB				
		286			
3113	TPUE				
		3096	3099		
2966	TRSET				
		2950	2960		
3211	TSCI				
		277			
3075	TSD2				
		3070			
3083	TSD3				
		3078			
2946	TSST				
		2931			
3065	TSYNC				
		NOT USED			
3094	TSYNCE				
		3073	3081	3086	3089
3091	TSYNCR				
		3062	3072	3080	3088
3627	VC01				
		3617	3622		
3643	VC02				
		3629			
3723	VC31				
		3804			
3724	VC32				
		3809			
3725	VC41				
		3828			
3726	VC42				
		3833			

4497	VCBT	247
4042	VCBUS	3922 4034
3905	VCCO	3756 3777 3813 3837 3902 3931 3946 3998 4016
3902	VCCP	3863 4032
4500	VCDB	251
3734	VCDHK	3676
4496	VCDT	250
3684	VCER1	3582
3690	VCER2	3657 3659 3661
3696	VCER3	3664 3666 3668 3670
3702	VCER4	3740 3743 3746
4056	VCER	3924
3677	VCGTF	3653
3712	VCHD	3580
3850	VCHE	NOT USED
3796	VCHFE	3675 3763 3768 3771 3784 3790
3891	VCKO	3897
3895	VCK1	3886
3930	VCK21	3883
3929	VCK2	3894 3898
3943	VCK3	3882 3890 3893
3960	VCK5	3941
3984	VCK6	3990
3988	VCK7	3979
3992	VCK8	3976 3987
4013	VCK9	3975 3983 3986

```
3867  VCK
                4038
3857  VCKFE
                3789
3573  VCLAS
                 164
3713  VCLD
                3581
4506  VCLOG
                 249
3921  VCM15
                3884  3885  3887  3889  3896  3977  3978  3980  3982  3989
3715  VCMA
                3655
3717  VCMOF
                3672  3674
3716  VCMT
                3662
4031  VCNK
                4011
3624  VCRET
                3592  3689  3695  3701  3707
4425  VCSDB
                 252
3719  VCST1
                3788
3720  VCST2
                3767
3721  VCST3
                3769  3770
4499  VCVC
                 248  1367
4051  VCVNB
                3923  3994  4010  4030
4468  XBUF
                 226
4283  XCCE
                4278
4240  XCER
                4229
4324  XCFT
                4323  4368
4367  XCFTE
                4368
4472  XCIWC
              NOT USED
4473  XCIWP
              NOT USED
4687  XERRO
              NOT USED
```

```
4688  XERR1
                NOT USED
4681  XERR
                  227
4682  XERRF
                NOT USED
4270  XFO1
                 4254
4296  XFO2
                 4280
4250  XFOW
                 4227
4368  XIMX
                 4253  4279
4385  XLOF1
                  207
4397  XLOF2
                  209
4379  XLON1
                  206
4391  XLON2
                  208
4257  XMCE
                 4252
4470  XOCW1
                NOT USED
4471  XOCW2
                NOT USED
4204  XRTC
                  172
4225  XSDC
                  174
4469  XSDW
                NOT USED
4215  XWDT
                  173
```

```
*****************************************************************************
*****************************************************************************

                 *****    ****        *        **     *****
                 *    *   *   *      **       * *    *
  ****     ****  *****    *****       *      *  *    *         ****     ****
                 *        *    *      *     ******   *
                 *        *    *      *        *     *
                 *        *****      ***       *     *****

*****************************************************************************
*****************************************************************************
-- 3 0 --        DATE  10-10-75    TIME  14.254       ID = G5   CS-F.0
```


GPO 907-801

TE 662
.A3
no.FHWA-RD-
76-68 BORROWER

DOT LIBRARY
00055535

FHWA

R&D